



SOFTWARE-DEFINED  
VIDEO PLATFORM

# Zixi Broadcaster User Guide

Software Version 14  
Document Version - DOC16-83-0029  
Copyright © Zixi 2021  
All Rights Reserved

# Legal Notice

This document contains proprietary and confidential material of Zixi LLC. Any unauthorized reproduction, use, or disclosure of this material, or any part thereof, is strictly prohibited. This document is solely for the use of Zixi employees and authorized Zixi customers.

The material furnished in this document is believed to be accurate and reliable. However, no responsibility is assumed by Zixi LLC. for the use of this document or any material included herein. Zixi LLC. reserves the right to make changes to this document or any material included herein at any time and without notice.

For more information visit: [www.zixi.com](http://www.zixi.com).

Copyright © Zixi 2021 Waltham, MA U.S.A.

All Rights Reserved.

*\* See back for additional licensing information*

# Table of Contents

Legal Notice.....	2
Introduction to Zixi Broadcaster .....	9
Zixi Software-Defined Video Platform (SDVP) .....	11
SDVP Architecture .....	13
Zixi Transport Stream Protocol .....	13
Supported Protocols .....	14
Transcoding Options .....	16
Recording Options .....	16
Multicast Options.....	16
Adaptive Options .....	16
VOD Options .....	16
Latency Considerations.....	16
Logging in to the Broadcaster .....	18
Activating Your License .....	19
Online Activation (License Key) .....	19
Offline Activation (License File) .....	19
Reloading the License .....	20
Upgrading the Product Version (Linux) .....	20
Viewing Status.....	22
General Status.....	22
Network Status .....	23
Viewing Current License Information .....	23
Metered License Monitoring .....	23
License Status Screen.....	25
Machine Information .....	26
Operating System Information .....	27
Network Cards (Linux only).....	27
Network Test.....	29
Installing the Zixi Video Acceleration Proxy.....	30

## Legal Notice

---

### Zixi Software-Defined Video Platform (SDVP)

Using the Zixi Video Acceleration Proxy for previewing a stream.....	33
Broadcaster Inputs.....	35
Adding Input Streams .....	35
Adding Push Streams .....	35
Adding Pull Streams .....	40
Adding UDP Streams.....	46
Adding Delayed Streams.....	51
Adding File Streams .....	53
Adding RTMP Streams .....	55
Adding RTSP Streams .....	60
Adding HLS Streams .....	64
Adding TS over HTTP (Pull) Streams .....	68
Adding TS over HTTP (Push) Streams.....	71
Adding RIST Streams .....	76
Adding NDI Streams .....	82
Viewing Existing Input Streams.....	87
Viewing Stream Statistics.....	88
Searching for Input Streams .....	91
Broadcaster Actions on Input Streams .....	91
Stopping/Starting the Input Stream .....	91
Editing Input Settings or Duplicating Input Streams.....	91
Deleting Input Streams from Broadcaster .....	92
Recording the Input Stream.....	92
Adding TR101 Analysis to the Input Stream .....	94
Transcoding an Input Stream.....	95
Add Output .....	100
Playing Input Streams with Players.....	102
Viewing Traceroutes .....	104
Broadcaster Actions on Multiple Inputs .....	104
Starting and Stopping Multiple Input Streams .....	104
Deleting Multiple Input Streams.....	105
Saving Multiple Input Streams.....	105

## Legal Notice

---

### Zixi Software-Defined Video Platform (SDVP)

Resetting Stats for Multiple Input Streams .....	105
Creating a Failover Group .....	105
Encrypting an Input Stream in Zixi Broadcaster .....	108
Decrypting an Input Stream in Zixi Broadcaster .....	109
Demuxing an MPTS Stream .....	110
Viewing TR101 Analysis .....	112
Stream Information Statistics Panel .....	113
P1/P2 Errors .....	116
Content Analysis .....	117
Enabling Content Analysis.....	118
Broadcaster Outputs.....	119
Adding Outputs .....	120
Adding UDP Outputs .....	120
Adding RTMP and RTMPS Outputs .....	124
Adding Pull Outputs .....	126
Adding Push Outputs .....	128
Adding TCP Outputs .....	131
Adding HTTP Push Outputs.....	134
Adding AWS Outputs .....	137
Adding RIST Outputs .....	139
Viewing Existing Outputs .....	144
Stats Pane.....	145
Searching for Output Streams .....	147
Performing Actions on Output Streams.....	147
Deleting Output Streams from Broadcaster .....	147
Switching Input .....	147
Viewing Traceroutes .....	147
Performing Actions on Multiple Outputs .....	149
Delete Marked Individual and Multiple Outgoing Stream .....	149
Start and Stop Marked Individual and Multiple Outgoing Streams .....	149
Save Marked Individual and Multiple Outgoing Streams .....	149
Reset Stats for Individual and Multiple Outgoing Streams .....	150

## Legal Notice

---

### Zixi Software-Defined Video Platform (SDVP)

Viewing Traceroutes .....	150
Using Output Templates .....	150
Broadcaster Network Bonding.....	154
Using Network Bonding for Hitless Failover .....	154
Broadcaster Output with Network Bonding .....	155
Adaptive Groups .....	156
Creating New Adaptive Groups .....	156
Adding Streams to Existing Adaptive Groups .....	159
Adaptive Groups Templates .....	160
Performing Actions on Adaptive Groups .....	163
Stopping/Starting an Adaptive Group .....	163
Editing an Existing Adaptive Group.....	164
Deleting and Adaptive Group .....	164
Play Dash .....	164
Play fMP4 .....	165
Play HLS.....	166
Set Preroll.....	167
VOD (Video on Demand).....	169
VOD Activation.....	169
VOD Files Location .....	169
Changing the Files folder location .....	171
Adding Files to VOD .....	171
Playing back VOD content.....	172
Indexing VOD content.....	173
Accelerated File Transfer .....	174
File Transfer Configuration .....	174
File Transfer Settings.....	174
Uploading Files for File Transfer .....	175
Downloading Files.....	176
Generating Accelerated URLs .....	177
Creating a New Folder .....	178
Renaming a File/Folder .....	179

## Legal Notice

---

### Zixi Software-Defined Video Platform (SDVP)

Deleting an Existing File/Folder .....	179
Transcoder .....	181
Creating New Profiles .....	181
Transcoding an Input Stream.....	187
Transcoding Templates.....	191
Server Settings .....	195
General.....	195
Network .....	199
Time .....	201
Live Protocols.....	202
Multicast Pool .....	205
File Transfer & VOD.....	206
Cluster .....	208
Zen Master .....	211
SSH Connections .....	213
Authorization .....	215
Logging .....	217
ASI .....	218
Reversing the directions of the ports .....	219
Threads.....	220
SNMP (Linux Only) .....	221
Viewing Events Log .....	223
Using the Matrix.....	224
Connecting Input to Output.....	224
Disconnecting Output from Input.....	225
Matrix Interface Elements .....	225
Changing Matrix View .....	226
Searching for Specific Streams.....	226
Preview Thumbnail Elements .....	226
Viewing Stream Details .....	227
Configuring SNMP .....	229
Activating SNMP .....	229

## Legal Notice

---

### Zixi Software-Defined Video Platform (SDVP)

Configuring SNMP Settings .....	229
Viewing MIB Information .....	231
SNMP Alerts/Notifications .....	231
Configuring Hitless Failover .....	233
Configuring HTTPS.....	235
Obtaining a CA-signed Certificate .....	235
Creating a Self-signed Certificate.....	235
Uploading the Certificate and Private Key.....	235
Configuring NDI Streams.....	237
NDI Stream Discovery .....	237
OS Support .....	239
Enabling Multicast Network Support.....	239
NDI config file location.....	239
Specifying the NDI Discovery Server .....	239
Configuring Transcoding Profiles for NDI Sources.....	240
Adding an NDI Input to Zixi Broadcaster .....	240
Creating an NDI Output .....	245
Configuring WebRTC Streams.....	249
Video and Audio Codec Support Options .....	249
Creating Transcoding Profiles for WebRTC.....	252
Creating a Video Profile for “Single Version” Transcoding.....	252
Creating an Audio Profile for “Single Version” Transcoding.....	252
Creating a Video Profile for “Multiple Version” Transcoding.....	253
Creating an Audio Profile for “Multiple Version” Transcoding .....	254
Creating a WebRTC Output.....	254
Playing and input stream with WebRTC player .....	256
Configuring MultiPath TCP.....	257
MultiPath Modes .....	257
MultiPath TCP Setup .....	259
Using an External Storage .....	260



# Introduction to Zixi Broadcaster

The Zixi Broadcaster is the central component in the Zixi Software-Defined Video Platform (SDVP). Zixi Broadcaster takes video streams from Zixi Feeder, and can process the stream, enabling transcoding, transmuxing, recording, and distribution in multiple bit rates and protocols to any device anywhere. The Zixi Broadcaster is an intelligent and versatile video distribution component that may reside on premise at a customer's site, on the customer's cloud, or offered by Zixi as a cloud service.

Zixi Broadcaster offers the following features:

- **Robust content delivery over IP** - communicates with Zixi Feeder and Zixi Receiver over UDP-based, video-optimized protocols on private and public IP networks for maximal quality at a predictable latency
- **Adaptive Bit Rate** - Dynamically adjusts stream rate to adapt to changing network conditions, using unicast or multicast, to meet specific application requirements
- **Transcoding** - transcodes to a variety of different profiles and bit rates
- **Format conversion** – supports Internet protocols: HLS/MPEG-DASH/RTMP/FLV/MPEG-TS over HTTP
- **Recording** - store streams as MPEG-TS files.
- **Time-shifting** – records the stream to delay its broadcasting
- **Secure and rapid file transfer** – accelerated and secure file transfers delivered at wire speed and accelerated HTTP delivery including optimized playback of HLS over UDP.
- **VOD** - stored files can be accessed on demand in multiple formats.
- **Clustering and load balancing** – supports cluster architectures to provide continuous uptime.
- **Transport Stream Analyzer** – MPEG-TS ETSI TR 101-290 analyzer (priorities 1 and 2)
- **Content Analysis** - analysis of the stream's audio and video content, including audio levels, audio silence detection, frozen video detection, and more.
- **Monitoring** - captures network and content specific statistical information in real-time.
- **Supports many to many and any to any** - gateway platform between UDP (unicast/multicast), Zixi protected stream (unicast/multicast), RTMP, or other supported formats.
- **Network bonding with hit-less fail-over** – reunites divided streams coming from multiple network channels into a single stream, enabling simultaneous load balancing between multiple networks and ensuring high availability of the stream.
- **Hitless failover** - hitless failover for inputs enables uninterrupted streaming when switching from one source to another. Zixi's new hitless failover feature is based on the SMPTE 2022-7 standard, which specifies “seamless” or hitless failover between binary-identical streams with synchronized RTP headers.
- **IFB Support** - a monitoring and cueing system for one-way communication from the director or assistant director to on-air talent or a remote location. IFB support has been

implemented in Zixi Broadcaster Pull and UDP outputs, allowing the decoder with a Zixi Receiver to send audio back to the encoder with a Zixi Feeder.

- **AES stream encryption and decryption** - UDP, File and RTMP input streams can be encrypted by Zixi Broadcaster using a fixed key (AES 128, AES 192, AES 256), which can be entered into the input settings or automatically generated. Zixi Broadcaster can also decrypt contribution streams using a matching key, or securely forward encrypted streams to receivers, without being able to see or analyze the traffic.
- **MPTS demuxing** - MPTS streams can be demuxed into single SPTS input streams for specific programs.
- **SCTE-35** - SCTE-35 markers found in the source input will be used to segment the stream and the #EXT-X-CUE-IN/#EXT-X-CUE-OUT tags will be added to the manifest.
- **Low-latency HLS** - Zixi Broadcaster supports low latency HLS delivery using chunked transfer encoding, which starts sending an HTTP response as chunks of data when they are ready before the complete response is available. Chunked transfer encoding can be use with transport stream (TS) segments and with fragmented MP4 (fMP4), which is also called Common Media Application Format (CMAF). For low latency HLS and DASH, Zixi Broadcaster must be used as an origin server.

## Zixi Software-Defined Video Platform (SDVP)

Zixi's proprietary Software-Defined Video Platform or SDVP is the tight integration of four components that enable broadcast-quality live video workflows to be intelligently and centrally provisioned, deployed, managed, and monitored using software and integrated devices, regardless of the underlying network infrastructure. employed by traditional broadcasters, OTT broadcast services, service providers to source, manage, and distribute live events and 24/7 live linear channels over any IP network, protocol, cloud provider, and/or edge device.

- **Protocols & Containers** – The SDVP accepts over 17 industry protocols and containers, advancing the market leading Zixi protocol while adding additional value to tangential offerings from other vendors, standards, and open-sourced initiatives. Most customers rely on the Zixi protocol for transport as it is congestion and network-aware and can dynamically adjust to varying network conditions and employ forward error correction techniques for error-free video delivery.
- **Video Solutions Stack** – Zixi's Video Solutions Stack provides essential tools and core media processing functions for broadcast-quality transport that corrects for packet loss and jitter. Patented Hitless Failover techniques provide 99.9999%+ reliable transport between sources and receivers. The Video Solutions Stack also provides broadcasters with enhanced tools to streamline workflows including protocol switching, analytics, live 4K transcoding and repackaging, recording, time shifted delivery, and auto-slating.
- **Intelligent Data Platform** – Zixi is constantly incorporating analytics, event correlation, AI and Machine Learning tools in its platform to aggregate a user's historical and real-time metadata and present it back to them to help better predict downtime, content quality, trends, and more in the form of alerts, notifications, and background monitoring.
- **ZEN Master** – With Zixi's ZEN Master control plane, cost-effectively manage and monitor complex deployments at scale and configure and orchestrate live broadcast channels across protocols and the Zixi Enabled Network. ZEN Master allows for control of the edge, network, and cloud with full telemetry visualization of network streams. It offers essential monitoring and management tools like workflow visualization, alerting, history, automation, scheduling, reporting, and root cause analysis across complex media supply chains.

### Seamless Protocol Switching Between 17 Protocols

Once users adopt the SDVP and install the software components, they can choose to send or receive live video streams using the Zixi protocol or one of the 16 supported protocols. With the SDVP, users can conduct protocol switching, handle up to 5Gb+ streams in 4K and now 8K

## Introduction to Zixi Broadcaster

---

### Zixi Software-Defined Video Platform (SDVP)

video, and change stream parameters without stopping the stream. SDVP allows for high availability with high redundancy and can run on premise or on any cloud.

### Hitless Failover, Analytics, and Bonding

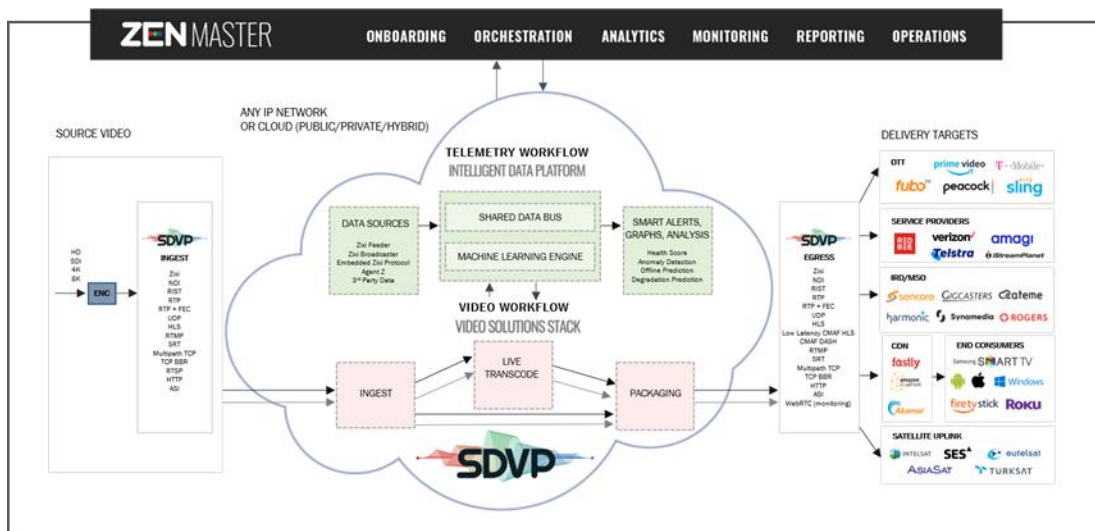
Users can take advantage of its Video Solutions Stack capabilities such as analytics, hitless failover and bonding across diverse ISPs to ensure high reliability and encrypt content with AES 256 encryption and DTLS for best-in-class security. The Video Solutions Stack's unique patented hitless failover techniques ensure reliability, and its robust analytics suite provides atomic-level telemetry on stream health that allows users to address any issues impacting QoS before the consumer's QoE is affected. The analytics solution employs machine learning techniques to provide a real-time estimated PSNR (ePSNR) with no reference video required, helping the user determine if content providers are meeting SLAs for contribution stream quality.

### Zen Master Control Plane

ZEN Master integration allows remote access to the UIs of Zixi-integrated devices and software across the Zixi Enabled Network for optimal monitoring and control from one centralized control plane. With the ZEN Master control plane, users have extensive visibility into and control over workflows that go across Zixi-enabled multi-party supply chains, with the ability to leverage alerting, extensive history graphs, automated toolsets, and scheduling and reporting capabilities that abstract complexity to reduce engineering requirements and allow media organizations to leverage operational resources, providing democratized access with geographic agility.

### SDVP Architecture

The Zixi SDVP architecture shows how a live linear broadcast or event is ingested through a Zixi Feeder (ZF) or Zixi embedded device running Zixi Feeder at acquisition. Zixi Feeder encapsulates the stream in the Zixi transport protocol and delivers point to point or point to multi-point over standard IP connections to Zixi Broadcaster (ZB). Zixi Broadcaster can be deployed on premise or in the cloud with the ability to monitor streams anywhere along the path. For management, processing and larger scale distribution capabilities, Zixi can support complex production workflows for live events, including transcoding, recording, and more, all in a clustered environment that supports reliability and scalability. The stream is distributed through a CDN, OTT, or endpoint device, running Zixi Receiver (ZR) to any device.



### Zixi Transport Stream Protocol

Zixi's transport stream protocol is a content and network-aware protocol that dynamically adjusts to varying network conditions and employs error correction techniques for error-free video streaming over IP. With minimum overhead to physical bandwidth, this dynamic mechanism provides low end-to-end latency, removes jitter, recovers and re-orders packets, smooths video delivery and regenerates video to its original form, all in real-time. Zixi delivers outstanding performance (at low predictable latency), superior reliability (no packet loss) and broadcast-grade video quality (SD, HD, and UHD) with no tradeoffs to delay, resolution or stutter.

Streaming from one Zixi-enabled device/server to another Zixi-enabled device/server protects the stream from quality degradations along the path. It enables the streaming of high-quality video over any distance, while overcoming the varying network conditions of the public Internet, where the amount of network errors, packet loss, jitter and out-of-order packets fluctuate "every second".

### Supported Protocols

The Zixi transport stream protocol can be also deployed on part of the path. For example, from a Zixi Feeder to a Zixi Broadcaster (Zixi protected path) and on to the end user through UDP or RTMP. In this case, the path that has challenging conditions (e.g., long distance or unstable wireless conditions) will be protected by Zixi and then delivered on to the end-user on a standard UDP or RTMP protocol.

Zixi's transport stream technology includes the following features for ultimate quality and security:

- Network sensing
- Dynamic de-jitter
- MPEG specific optimizations
- Z-ARQ error recovery
- Z-FEC - Dynamic content aware forward error correction
- Active multi-path error recovery
- Adaptive bitrate over UDP, unicast or multicast
- Rate control and congestion avoidance
- 256-bit AES transport encryption

## Supported Protocols

Input:

- Zixi protected transport
- MPEG-TS over UDP and/or RTP with SMPTE-2022 and/or
- RTMP pull from CDNs and/or other sources
- RTMP push
- RTMPS
- RTSP input
- RIST (TR-06-1)
- NDI
- Transport stream files from local file system
- SRT (configured via Zen Master or Zixi API)

### Supported Protocols

#### Output:

- Zixi protected transport
- MPEG-TS over UDP and/or RTP with SMPTE-2022 and/or RIST (TR-06-1)
- RTMP and RTMPS push to CDNs and/or other media servers;
- Apple HTTP Live Streaming (HLS)
- Adobe HTTP Dynamic Streaming (HDS)
- FLV over HTTP (HTTP pseudo-streaming)
- MPEG-DASH (DASH264 profile)
- SHOUTcast
- NDI
- Transport stream files to local file system
- Re-multiplex output streams to strict CBR for extensive compatibility with Integrated Receiver- Decoders (IRDs)
- SRT (configured via Zen Master or Zixi API)

## Transcoding Options

Zixi Broadcaster can transcode a single input stream using multiple profiles. The transcoder supports the following options:

- Video Decoding – MPEG2, H.264/AVC, H.265/HEVC, Raw
- Video Encoding – MPEG2, H.264/AVC, H.265/HEVC, Raw
- Audio Decoding – AAC, MPEG-1 Audio Layer I / II / III, MPEG-2 Audio Layer I / II / III, AC-3 (Dolby Digital), Opus.



For AC-3, which can contain up to six discrete channels of sound, the transcoder only supports the left and right channels and will ignore the other channels.

- Audio Encoding – AAC, AAC-HE (High Efficiency) and AAC-HEv2; Pass through (no encoding) – any audio codec (MPEG1/2 / AAC / AC-3, E-AC-3 (Dolby Digital Plus), Opus.

## Recording Options

Live stream inputs can be stored as MPEG-TS files to a local disk drive or mounted drive.

## Multicast Options

Multicast streams can be received or be sent by the Zixi Broadcaster. Zixi Broadcaster serves as a gateway between Unicast and Multicast streams, while allowing reliable streaming over multicast.

## Adaptive Options

Zixi Broadcaster can create an adaptive group, subsequently editing, recording, or playing it in conjunction with DASH, HLS, HDS or Zixi. You can also set a Pre-roll video/screen that will be played before the live streaming begins.

## VOD Options

Zixi Broadcaster supports VOD (Video on Demand) playback of stored files in multiple formats such as TS, MP4, HLS, FLV.

## Latency Considerations

Latency defines the delay in which the video stream will be delivered to the endpoint. This delay is essentially a buffer that is used for additional processing, thereby improving the quality of video stream. The latency should be considered based on the importance of immediacy. For example, if the video requires interaction (e.g., between a reporter in the field and an anchor in



### Latency Considerations

the studio), the latency should be set to a minimum so that the interaction does not suffer from any delay.

Latency can vary from tens of milliseconds to several seconds.

Zixi protocol can start providing benefit from one frame (30ms), while increasing the latency will provide more robustness for the error-recovery.

# Logging in to the Broadcaster

### ➔ To log in to Zixi Broadcaster:

1. In your web browser, navigate to <http://localhost:4444>.  
A user authentication window opens.
2. In the **User Name** field, enter '**admin**'.
3. In the **Password** field, enter '**1234**'.
4. Click **OK**.  
The Broadcaster UI opens on the **Status** page.

# Activating Your License

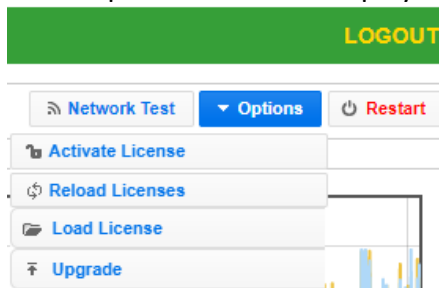
In order to start using Zixi Broadcaster, first activate your license. Zixi Broadcaster and its optional features are activated by license files. Before using Zixi Broadcaster, and when adding optional features, **activate** the appropriate license.

## Online Activation (License Key)

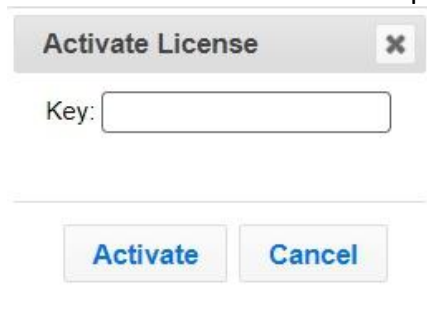
The following procedure requires connectivity to the Internet. If you are not able to connect, you can activate the Broadcaster using the procedure described in [OFFLINE ACTIVATION \(LICENSE FILE\)](#).

### To Activate the License online using a License Key:

1. In the **Status** page of the Zixi Broadcaster UI, click **Options** in the menu bar. The drop-down menu is displayed.



2. Click **Activate License**. The **Activate License** window appears.



3. In the **Key** field, enter your Activation Key.
4. Click **Activate**. The Zixi Broadcaster is enabled.

## Offline Activation (License File)

If you want to activate your license without an internet connection, you can upload a License File. To obtain a license file, please contact *Zixi Support* at [support@zixi.com](mailto:support@zixi.com).

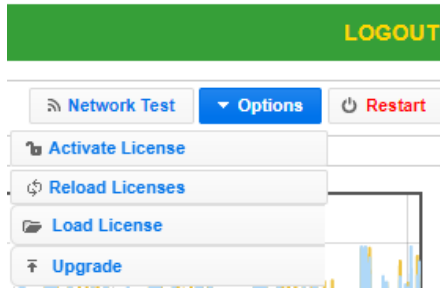
## Activating Your License

---

### Reloading the License

#### ➔ To Load the License File:

1. In the **Status** page of the Zixi Broadcaster UI, click **Options** in the menu bar. The drop-down menu is displayed.



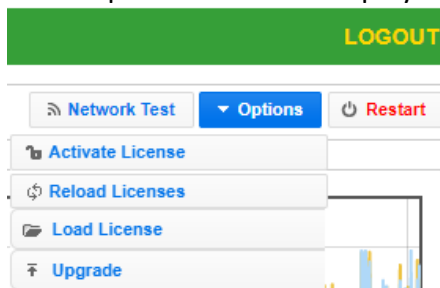
2. Click **Load License**.  
The directory of the local drive opens.
3. Navigate to the License File and click **Open**.  
A prompt window instructs you to Reload the license to activate the license.
4. Click again on **Options** and select **Reload Licenses** from the dropdown menu.  
The Zixi Broadcaster is enabled.

## Reloading the License

In the event that your license expires, you must re-enter a new Zixi license key.

#### ➔ To reload the license:

1. In the **Status** page of the Zixi Broadcaster UI, click **Options** in the menu bar. The drop-down menu is displayed.



2. Click **Reload License**.  
The Zixi Broadcaster is enabled.

## Upgrading the Product Version (Linux)

When a new version of Zixi Broadcaster becomes available, you will receive a new installation archive file (.tar.gz file) from your Zixi representative to upgrade your Zixi Broadcaster software. It is recommended to perform the installation within a "maintenance window" and notify end users that the streaming will be momentarily interrupted. The upgrade process may last a few minutes after which the streams will be automatically resumed.

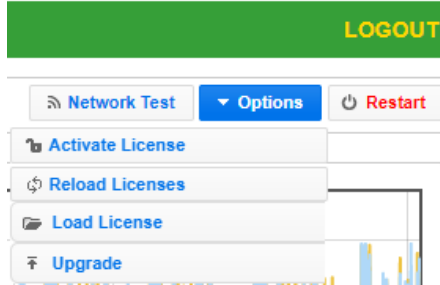
## Activating Your License

---

### Upgrading the Product Version (Linux)

#### ➔ To upgrade the version:

1. In the **Status** page of the Zixi Broadcaster UI, click **Options** in the menu bar. The drop-down menu is displayed.



2. Click **Upgrade**.
3. Select the relevant installation archive file (.tar.gz file) and click **Open**. The installation process will begin. The process may take up to a few minutes. During the process, the streams and the management UI will not be available. After the upgrade, all streams will be automatically resumed.

# Viewing Status

The Status screen displays aggregated statistics on all the input and output streams and a graph representing the network utilization of active streams over time.

#### ➔ To View the Status of the Streams:

In the Zixi Broadcaster **Status** page, the **General** area, **Network Status** area, **License Information** area, **Machine Information** area and **Operating System** area are displayed.

## General Status

---

---

General	
<b>ID</b>	The Zixi Broadcaster ID number
<b>Host ID</b>	Displays the ID of Zixi Broadcaster host.
<b>Start Time</b>	Displays the time that the Zixi Broadcast began to actively stream content.
<b>Up Time</b>	The duration of time that the broadcast is running.
<b>Inputs</b>	The number of current active inputs
<b>Outputs</b>	The number of current active outputs
<b>Remote In [kbps]</b>	The current aggregate input bitrate from remote sources – Zixi Push/Pull (in Kbps)
<b>Local In [kbps]</b>	The current aggregate input bitrate from local sources – UDP streams (in Kbps)
<b>Remote Out [kbps]</b>	The current aggregate output bitrate to remote destinations (in Kbps)
<b>Local Out [kbps]</b>	The current aggregate output bitrate to local destinations (in Kbps)
<b>HTTP Out [kbps]</b>	The current aggregate output bitrate for HTTP streams - FLV/RTMP/HLS/ MPEG-TS over HTTP (in Kbps)
<b>kBytes Sent</b>	The cumulative traffic transmitted (in Kbps)
<b>kBytes Received</b>	The cumulative traffic received (in Kbps)
<b>RTMP Server</b>	Active – the RTMP server is enabled (See Settings > Live Protocols > RTMP Server section) Off – the RTMP server is disabled

---

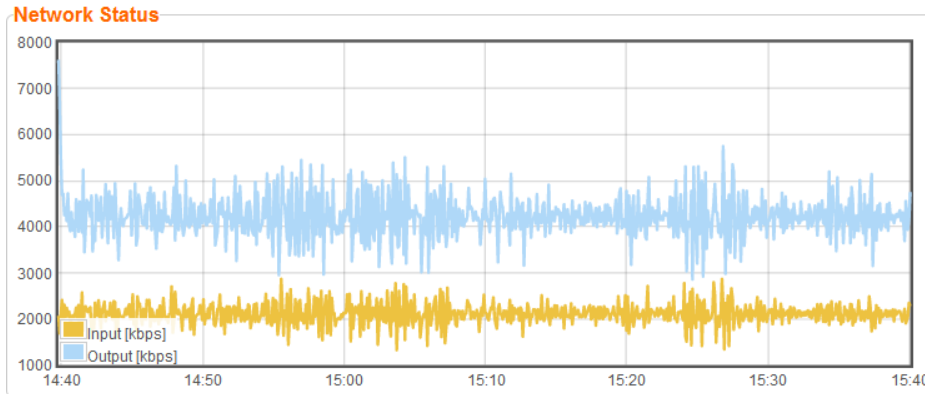
---

## Viewing Status

### Network Status

## Network Status

The network status graph displays the total input/output bitrate during the last hour. The changes in the amplitude of both the input (gold) and output (blue) presented in 10-minute intervals.

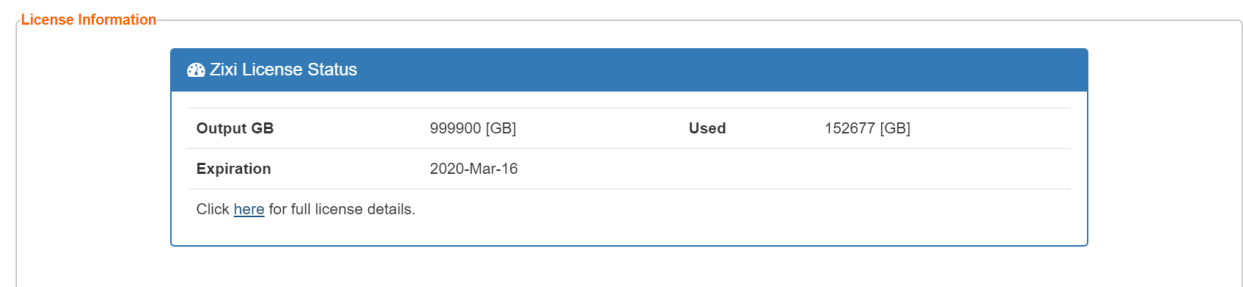


## Viewing Current License Information

You can view the current license information in the **License Information** section.

## Metered License Monitoring

If you have a metered license (based on usage), you can view the license conditions and monitor your current usage status under the License Information section on the main screen.



The section includes the following elements:

Element	Description
Output GB	The total amount of output traffic in GB per month that is allowed in the license.
Expiration	License expiration date.
Used	The actual usage this month so far in GB.

## Viewing Status

---

### Viewing Current License Information

To view additional license terms and usage statistics, click the link at the bottom of the Zixi License Status box.

License Information

**Zixi License Status**

<b>Output GB</b>	999900 [GB]	<b>Used</b>	152677 [GB]
<b>Expiration</b>	2020-Mar-16		
Click <a href="#">here</a> for full license details.			

The License Status Screen is displayed.



# Viewing Status

## Viewing Current License Information

### License Status Screen

#### Zixi License Status

##### Application

Product	Key
Broadcaster_new	demo-f14c-c546-6784

##### Meters

Meter	Expiration
Output GB	999900 [GB] 16-Mar-2020
Available	999900 [GB] Used 152677 [GB]
Expires	2020-Mar-16

##### Limits

	Input	Output
Bitrate	Unlimited	Unlimited
Zixi Pull	Unlimited	Unlimited
Zixi Push	Unlimited	Unlimited
RTMP Pull	Unlimited	Unlimited
RTMP Push	Unlimited	Unlimited
UDP	Unlimited	Unlimited
ASI	Unlimited	Unlimited
Transcoders	999	

##### Features

	Supported
HLS, HDS, FLV, MPEG-DASH, VOD & RTMP	<input checked="" type="checkbox"/>
Receivers & UDP	<input checked="" type="checkbox"/>
SMPTE 2022 FEC for RTP	<input checked="" type="checkbox"/>
TR101 Stream analysis	<input checked="" type="checkbox"/>
File transfer & Zixi accelerated HLS/HDS	<input checked="" type="checkbox"/>
Cluster	<input checked="" type="checkbox"/>
Matrix	<input checked="" type="checkbox"/>
Bonding	<input checked="" type="checkbox"/>
Time shifting	<input checked="" type="checkbox"/>
x264 Transcoding	<input type="checkbox"/>

##### Daily Usage

Host: All

Date	Usage (GB)
2019-03-26	0
2019-04-01	0
2019-04-03	0
2019-04-05	0
2019-04-07	0
2019-04-09	0
2019-04-11	0
2019-04-13	0
2019-04-15	0
2019-04-17	0
2019-04-19	0
2019-04-21	0
2019-04-23	0
2019-04-25	0
2019-04-27	0
2019-05-01	0
2019-05-03	0
2019-05-05	0
2019-05-07	0
2019-05-09	0
2019-05-11	0
2019-05-13	0
2019-05-15	0
2019-05-17	0
2019-05-19	15,000,000
2019-05-21	40,000,000
2019-05-23	38,000,000
2019-05-25	25,000,000
2019-05-27	0

##### Daily Bitrate

Host: All

Date	Bitrate
2019-03-26	0
2019-04-02	0
2019-04-09	0
2019-04-16	0
2019-04-23	0
2019-04-30	0
2019-05-07	0
2019-05-14	0
2019-05-21	4,500
2019-05-28	0

## Viewing Status

### Machine Information

The screen includes the following elements:

Element	Description
<b>Application</b>	The product version and license key.
<b>Meters</b>	<ul style="list-style-type: none"><li>• Output GB - The total amount of output traffic in GB per month that is allowed in the license.</li><li>• Expires - license expiration date.</li><li>• Available - the total amount of traffic in GB per month that is allowed in the license.</li><li>• Used - the actual usage this month so far in GB..</li></ul>
<b>Limits</b>	License usage restrictions. Unlimited - no restrictions.
<b>Features</b>	List of enabled/disabled features.
<b>Daily Usage</b>	Daily traffic statistics. Click the Host drop-down menu to select a specific host.
<b>Daily Bitrate</b>	Daily bitrate statistics. Click the Host drop-down menu to select a specific host.

## Machine Information

This pane shows information about the machine on which the Broadcaster is installed.

### Machine Information

**System Model:** Xen HVM domU  
**Processor Model:** Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz  
**Processors:** 1  
**Logical Cores:** 2  
**Installed RAM:** 4 GB  
**GPU(s):** Not Found

The pane includes the following data:

Parameter	Description
<b>System Model</b>	The model of the machine on which the Broadcaster is installed.
<b>Processor Model</b>	The model and specifications of the machine's processor.
<b>Processor</b>	The number of CPUs installed on this machine.

## Viewing Status

### Operating System Information

<b>Logical Cores</b>	The number of logical cores per processor (for each processor).
<b>Installed RAM</b>	The size of the RAM installed on the machine.
<b>GPU(s)</b>	The number of GPUs. If no GPUs - Not Found.

## Operating System Information

This pane shows information about the operating system running on the machine.

### OS Information

```
OS: Amazon Linux release 2 (Karoo)
Build: 4.14.88-88.73.amzn2.x86_64
Architecture: x86_64
```

The pane Includes the following data:

Parameter	Description
<b>OS</b>	The operating system running on the machine.
<b>Build</b>	The operating system build (version) number.
<b>Architecture</b>	The system architecture of the operating system, number of bits.

## Network Cards (Linux only)

This pane shows information about the network cards running on the machine. Each line represents another network card.

### Network Cards

```
NIC      Link Speed  Duplex  IPs
eth0    Full        Full    172.31.17.149
```

The pane Includes the following data:

Parameter	Description
<b>NIC</b>	The name of the Network Interface Controller (a.k.a network card).
<b>Link Speed</b>	The link speed capability of the network card.
<b>Duplex</b>	Full - full duplex Half - half duplex

## Viewing Status

---

Network Cards (Linux only)

---

---

**IPs**

The IP of the network card.

---

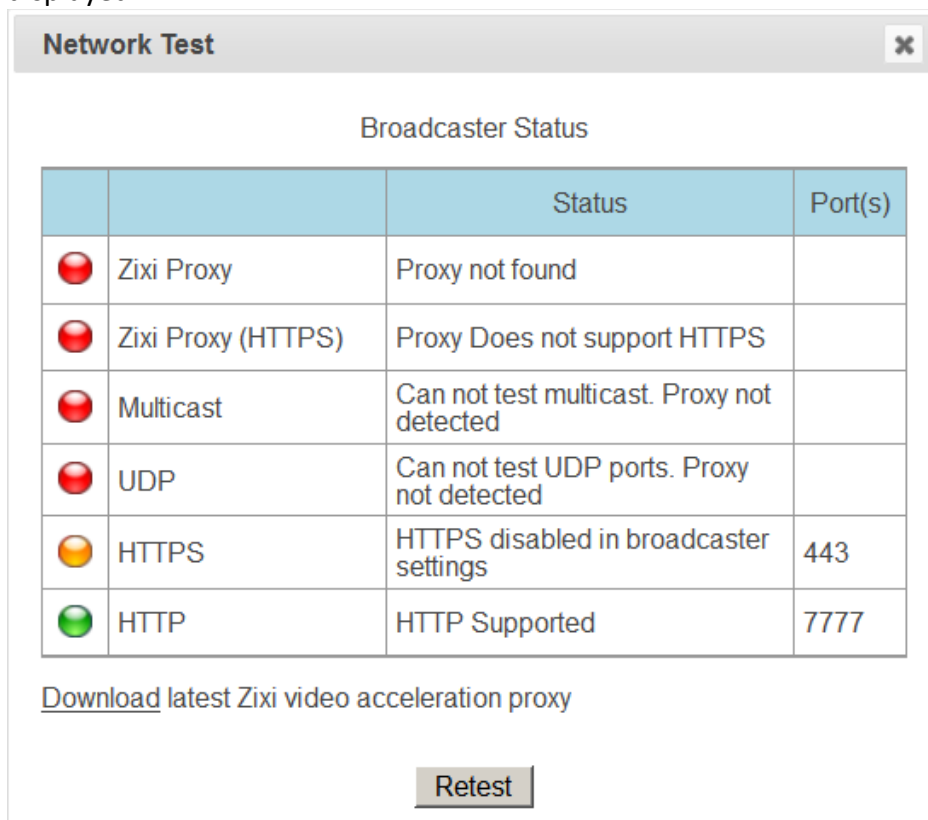
---

# Network Test







You can test the Zixi Broadcaster status, the status of the http or https connection with the relevant port, and the status of the Zixi Video Acceleration Proxy component. This can be accomplished remotely after the installation of the Broadcaster and subsequently any time after that.

### ➔ To Test the Network:

1. In the Zixi Broadcaster **Status** page, click . The **Network Test** screen is displayed:



The screenshot shows a window titled "Network Test" with a close button. Inside, there is a section titled "Broadcaster Status" containing a table with the following data:

		Status	Port(s)
	Zixi Proxy	Proxy not found	
	Zixi Proxy (HTTPS)	Proxy Does not support HTTPS	
	Multicast	Can not test multicast. Proxy not detected	
	UDP	Can not test UDP ports. Proxy not detected	
	HTTPS	HTTPS disabled in broadcaster settings	443
	HTTP	HTTP Supported	7777

Below the table is a link: [Download latest Zixi video acceleration proxy](#). At the bottom of the window is a **Retest** button.

The Broadcaster Status table lists the various components of the Zixi Broadcaster with the relevant ports and status (description) and the following health indicators:

- **Green** – component has been tested and was found available.
  - **Yellow** – component has been disabled.
  - **Red** – component was not found or is not available.
2. Click **Retest** to run an additional test on all the server's current connections and their relative ports.

The following tests will be displayed:

## Network Test

---

### Installing the Zixi Video Acceleration Proxy

- **Zixi Proxy** – checks that the Zixi Video Acceleration Proxy is running on the local machine and is listening through port 4500 (HTTP).
- **Zixi Proxy (HTTPS)** – checks that the Zixi Video Acceleration Proxy is running on the local machine and is listening through port 4501 (HTTPS).
- **Multicast** – checks if the multicast feature is enabled.
- **UDP** – checks if Zixi Broadcaster can be accessed through UDP (port 2077).
- **HTTP** – checks is Zixi Broadcaster can be accessed through HTTP (HTTP server port 80).
- **HTTPS** - checks is Zixi Broadcaster can be accessed through HTTPS (HTTPS server port 8080)

## Installing the Zixi Video Acceleration Proxy

The Zixi Video Acceleration Proxy is a software component, which is installed on the local machine, that enables pulling a stream from Zixi Broadcaster over HTTP, while using the Zixi protocol. The Zixi Video Acceleration Proxy is mainly used to preview a stream to assess its quality, by playing it on a local player. VLC video player has another version of the proxy, called Zixi VLC Plugin, which offers the same functionality. For more information, see [Zixi VLC Plugin Setup Guide](#).

The Zixi Video Acceleration Proxy component is installed on the local machine, while Zixi Broadcaster can be running on another machine (e.g., in the cloud) or can be running on the same machine. The Proxy uses port 4500 (HTTP) or 4501 (HTTPS) to listen to the stream.

The Zixi Video Acceleration Proxy is also required when using Zixi Broadcaster to transfer large files. In this case, the component allows both uploading and downloading the files. For more information, see [Accelerated File Transfer](#).

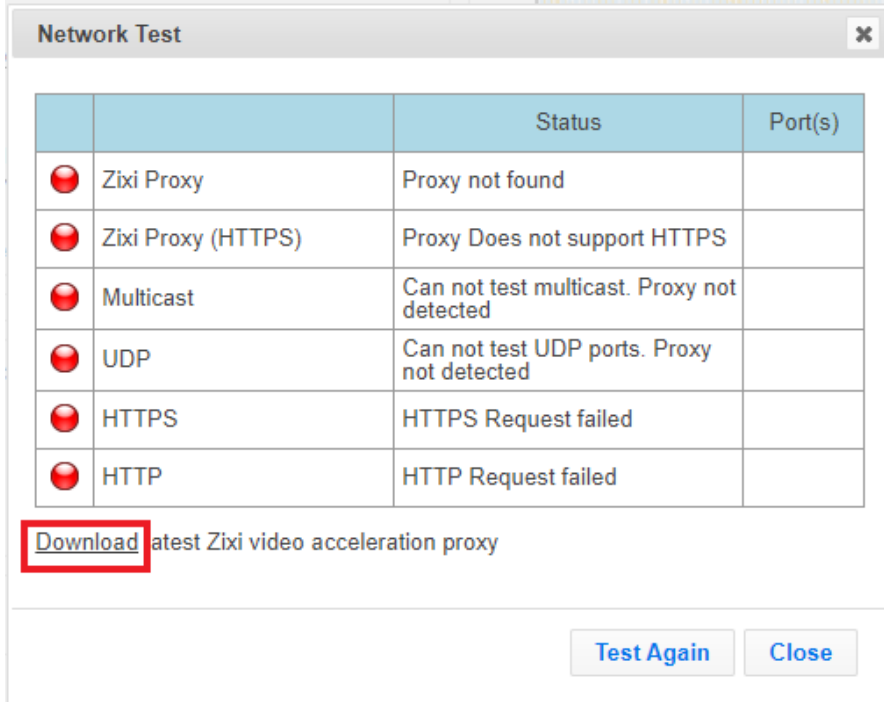
### ➔ To Install the Zixi Video Acceleration Proxy:

1. In the Zixi Broadcaster **Status** page, click . The **Network Test** screen is displayed.

## Network Test

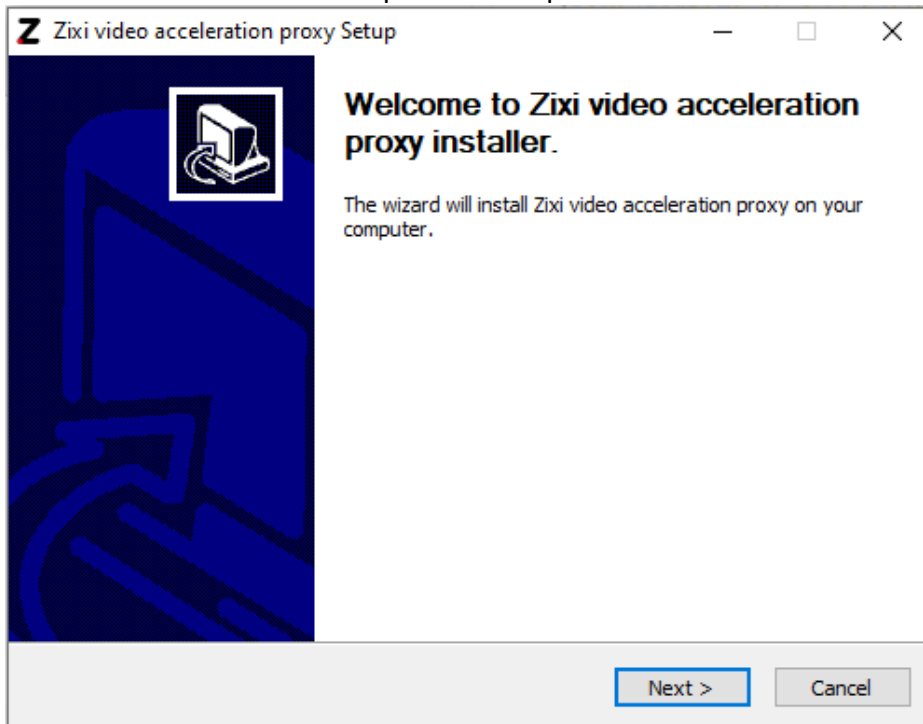
### Installing the Zixi Video Acceleration Proxy

2. On the **Network Test** screen, click the **Download** link below the table.



The setup file will be downloaded locally.

3. Run the downloaded file to open the setup wizard.

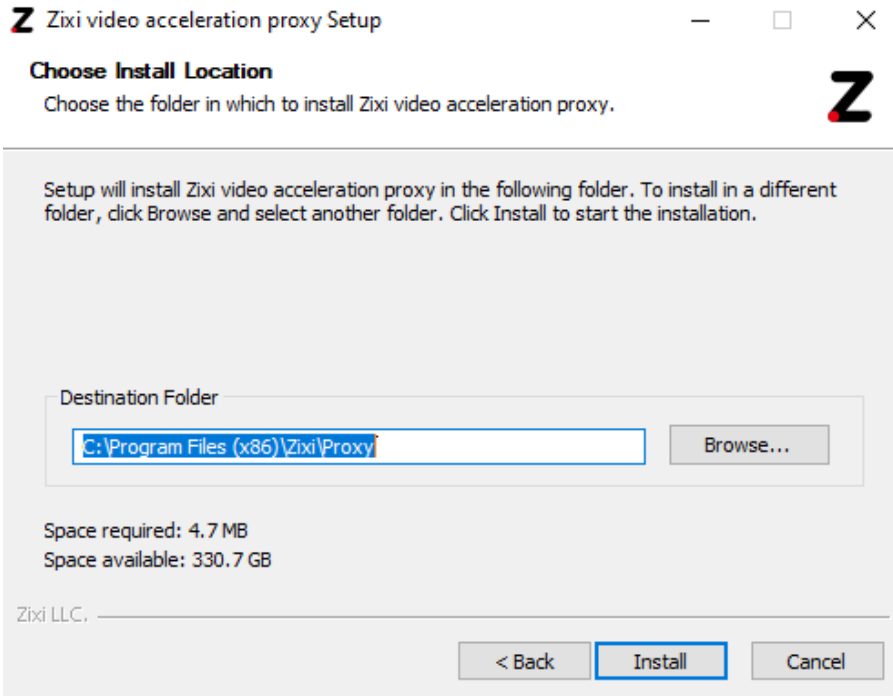


4. Click **Next**.

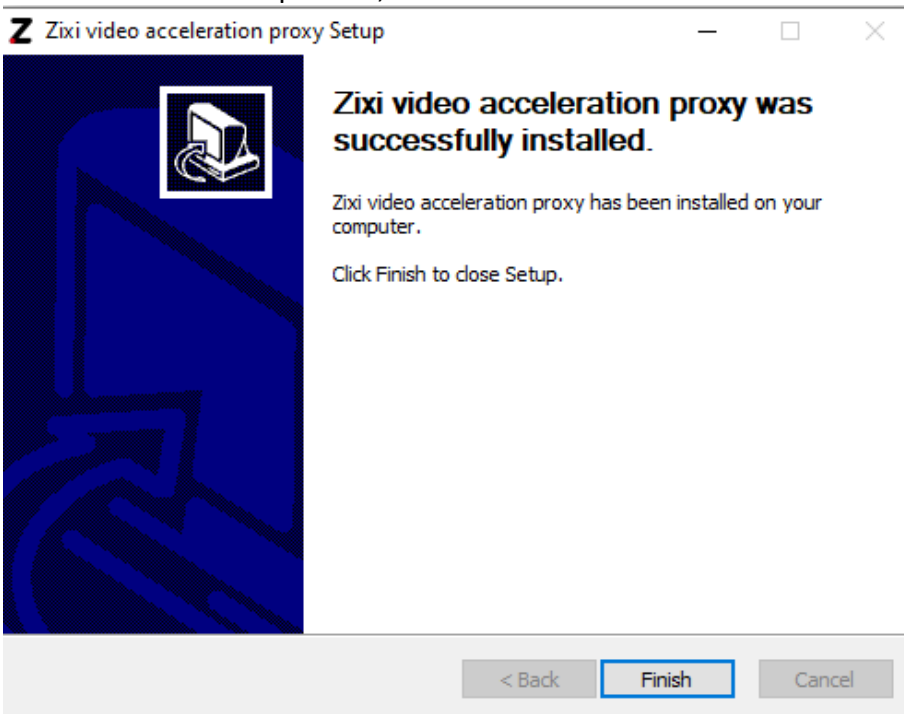
## Network Test

### Installing the Zixi Video Acceleration Proxy

5. Select a destination folder to install.



6. Click **Install**.
7. After the installation process, click **Finish**.

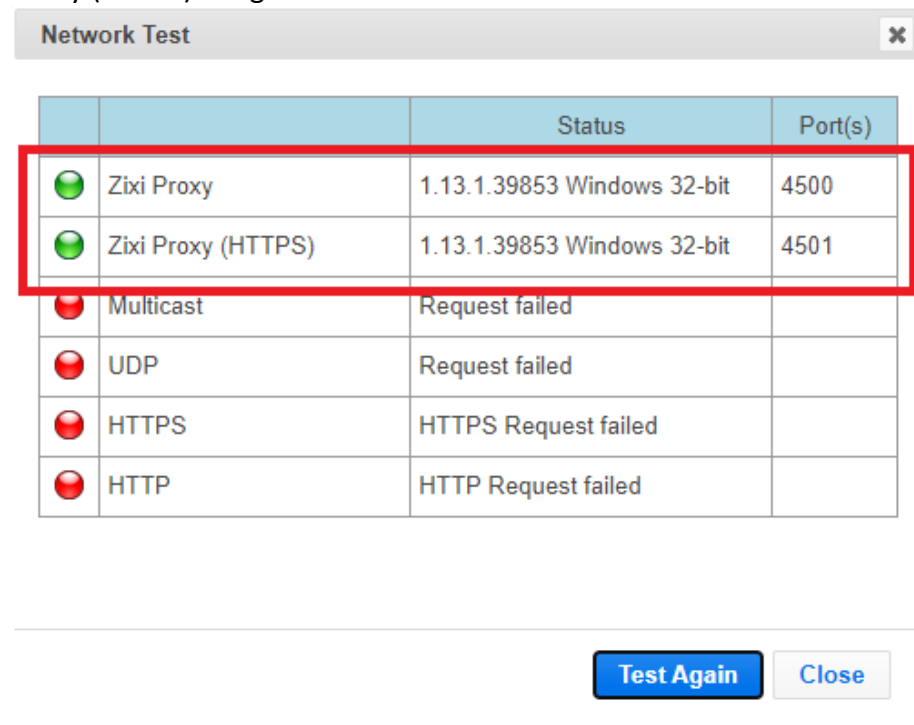










## Network Test

Using the Zixi Video Acceleration Proxy for previewing a stream

If you will run the network test again, you should be able to see that the Zixi Proxy and Zixi Proxy (HTTPS) are green.



		Status	Port(s)
	Zixi Proxy	1.13.1.39853 Windows 32-bit	4500
	Zixi Proxy (HTTPS)	1.13.1.39853 Windows 32-bit	4501
	Multicast	Request failed	
	UDP	Request failed	
	HTTPS	HTTPS Request failed	
	HTTP	HTTP Request failed	

**Test Again** **Close**

## Using the Zixi Video Acceleration Proxy for previewing a stream

The Zixi Video Acceleration Proxy can be used for previewing a stream through a video player. The locally installed Proxy can be accessed through the following IP address – 127.0.0.1. The following URL - 127.0.0.1:4500/version.json will provide a response that includes the version and build numbers of the Proxy.

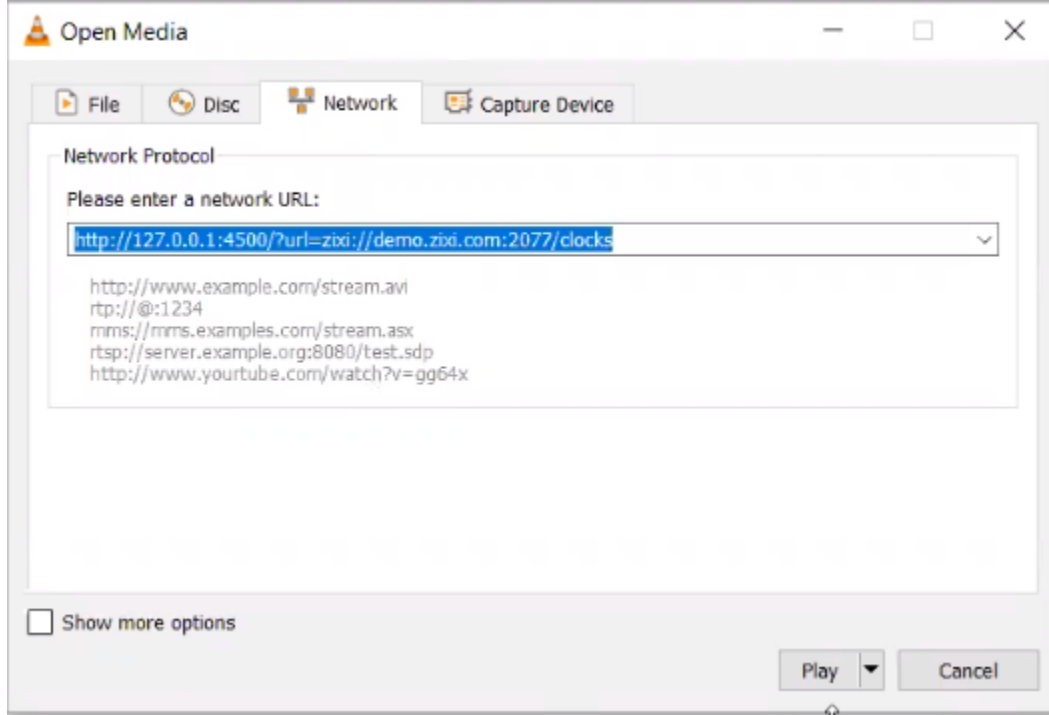
### ➔ To playback a stream using the Zixi Video Acceleration Proxy:

1. In a web-connected video player, enter the following network URL:  
[http://127.0.0.1:4500/?url=<broadcaster\\_ip>:2077/<stream\\_name>](http://127.0.0.1:4500/?url=<broadcaster_ip>:2077/<stream_name>).

## Network Test

Using the Zixi Video Acceleration Proxy for previewing a stream

The video will be streamed to the player through the locally installed Proxy.



# Broadcaster Inputs

You can add input streams to Zixi Broadcaster from one of the following sources:

- **Zixi Feeder (Push)** – the Zixi feeder pushes the stream to the Zixi Broadcaster. Since both the Zixi Feeder and the Zixi Broadcaster use the Zixi protocol, the stream is protected by the Zixi protocol.
- **Another Zixi Broadcaster (Pull)** – the Zixi Broadcaster can pull a stream from another Zixi Broadcaster. Since both components use the Zixi protocol, the stream can be protected by the Zixi protocol.
- **UDP streams** - the source is a UDP or RTP stream, RTP + SMPTE 2022 FEC over unicast or multicast.
- **Delayed** - input stream will be stored on the local disk, while a variant of the input is created and delayed according to the specified duration.
- **Local files (for VOD)** – the source in this case is a local transport stream file from your local computer or local network.
- **RTMP streams** - The source is an RTMP stream, pushed into the Zixi Broadcaster or pulled by it.
- **RTSP** - The source is an RTSP stream, pushed into the Zixi Broadcaster, or pulled by it.
- **HLS** - The source is an HLS stream that is pulled by the Zixi Broadcaster.
- **HTTP Pull/Push (TS over HTTP)** - The source is a Transport Stream over HTTP, which is pushed into Zixi Broadcaster or pulled by it.
- **RIST** – The source is a RIST stream using Simple or Main Profile over unicast or multicast.
- **SRT** – The source is an SRT stream and is configured via Zen Master or Zixi API only.

## Adding Input Streams

The Zixi Broadcaster enables you to add various types of streams. When you create a new stream, select the stream type and then fill in the fields that are applicable for that stream type.

### Adding Push Streams

Typically, a Push Stream is transmitted from a Zixi Feeder. In this case, the stream is initiated by the Zixi Feeder while the Zixi Broadcaster remains in “listening mode” until the stream is initiated.

If the stream is an MPTS stream and you want to demux it into separate SPTS input streams, create the stream, by following the instructions below, and then follow the instruction in the [Demuxing an MPTS Stream](#) section.

#### ➔ To Add a Push Stream:

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.

## Broadcaster Inputs

### Adding Input Streams

2. In the **Menu** bar, click **New Input**. The **Add a new input stream** window is displayed.

The screenshot shows the 'Add new input stream' dialog box. It features a title bar with a close button. The main area contains several input fields and checkboxes. The 'Stream ID' field is empty. The 'Max Outputs' dropdown is set to 'Unlimited'. The 'Show in Matrix' checkbox is checked. Below these are radio buttons for 'Push' (selected), 'Pull', 'UDP', 'Delayed', 'File', 'RTMP', 'RTSP', 'HLS', 'TS over HTTP', and 'RIST'. A section titled 'Stream parameters' includes 'Password' (empty), 'Latency [ms]' (set to 'Remote configuration'), 'Point to point' (unchecked), and 'ID of high priority source' (set to 'None'). Below this are checkboxes for 'Enable time shift', 'Transmit as multicast', 'Decrypt stream', 'Enable encryption', and 'Recording parameters', all of which are unchecked. At the bottom right are 'Ok' and 'Cancel' buttons.

3. Enter the unique **Stream ID** for the Broadcaster.



Note: this must be identical (case sensitive) to the stream name configured in the Zixi Feeder.

4. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.



This setting can be modified without interrupting an existing stream.

5. Select the **Show in Matrix** checkbox to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
6. Select **Push** for this input stream.
7. Specify the **Stream Parameters** (see Input Stream Parameters Table below).
8. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameter Table below). The stream will be recorded and broadcasted after the specified delay. The time shift feature is typically used to broadcast a live event across time zones (due to the time differences between locations).
9. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameter Table).
10. If you want to record the stream, you can configure the **Recording Parameters** (see Input Stream Parameter Table).
11. You can decrypt an encrypted stream as it enters Zixi Broadcaster. To decrypt a stream, select the **Decrypt Stream** checkbox and fill in the parameters as describes in the Input



## Broadcaster Inputs

### Adding Input Streams

Stream Parameter Table below. For more information, see [Decrypting an Input Stream in Zixi Broadcaster](#).



- Click **OK**. The “Input added” message appears on the top of the screen. The Push Stream from the Zixi Feeder is added to the Zixi Broadcaster and now appears in the list of input streams available in the Zixi Broadcaster input UI.

### Input Stream Parameters Table

Parameter	Description
<b>Stream Parameters</b>	
<b>Password (Optional)</b>	<p>If desired, enter a password string to use for authentication (must be identical to the string configured on the Feeder)</p> <p> This setting can be modified without interrupting an existing stream.</p>
<b>Latency [ms]</b>	<p>Specify the maximal latency that will override the latency that was defined in the Feeder (in ms). Default: Remote configuration (i.e. use the latency was set in the Feeder). (See <a href="#">Latency Considerations</a>)</p> <p> This setting can be modified without interrupting an existing stream.</p>
<b>Point to point only</b>	<p>Selecting this checkbox transmits the stream directly (point-to-point) from the Feeder to the Receiver, while only delivering messaging to the Broadcaster. In case there is a problem with the P2P connection, it will re-route the transmission through the Zixi Broadcaster.</p>
<b>Disable P2P Fallback</b>	<p>Selecting this checkbox will allow <u>only</u> P2P (disabling the option to re-route traffic though the Zixi Broadcaster)</p>
<b>ID of high priority source</b>	<p>Specify the ID of the Zixi Feeder for which you want to grant priority. The ID of the Zixi Feeder can be found in the <b>ID</b> field on the Zixi Feeder’s <b>Status</b> screen. Upon connection of the Zixi Feeder with this ID, all other existing sources will be disconnected.</p>
<b>Enable time shift</b>	<p>Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a>) .</p>
<b>Maximum delay [hh:mm:ss]</b>	<p>Specify the time of the delayed transmission of this stream. Maximum: 24 hours.</p>

## Broadcaster Inputs

### Adding Input Streams

<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default, Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling <b>Multicast Pull</b> in the <b>Settings</b> .
<b>Multicast Only</b>	Selecting this checkbox will force the transmission of this stream only in multicast.
<b>Decrypt stream</b>	Select this checkbox to decrypt the Input stream. Please note that the stream will not be encrypted as it passes through the Zixi path. For more information, see <a href="#">Decrypting an Input Stream in Zixi Broadcaster</a> .
	Specify the type of encryption that is being used in the stream.
<b>Decryption type (for Decrypt stream enabled)</b>	 This setting can be modified without interrupting an existing stream.
	Specify the key for decryption.
<b>Decryption key (for Decrypt stream enabled)</b>	 This setting can be modified without interrupting an existing stream.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a> .
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).
<b>Encryption key (for Encryption enabled)</b>	Click <b>Generate</b> to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.
<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.
	Select the radio button for the desired storage type and then fill in the relevant parameters.
<b>Destination Type</b>	<ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see <a href="#">Using an External Storage</a> section.</li><li>• <b>Record to S3</b> – the file is saved to your AWS S3 Bucket.</li></ul>

## Broadcaster Inputs

---

### Adding Input Streams

---

---

#### **Record to Disk Parameters**

---

---

<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
---------------------------	---

---

---

<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours

---

---

<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.
--------------------	--

---

---

#### **Record to S3 Parameters**

---

---

## Broadcaster Inputs

### Adding Input Streams

<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>URL</b>	Specify the URL of the S3 bucket.
<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.

### Adding Pull Streams

Typically, a Pull Stream is transmitted from another Zixi Broadcaster. In this case, the stream is initiated by your Zixi Broadcaster.

If the stream is an MPTS stream and you want to demux it into separate SPTS input streams, create the stream, by following the instructions below, and then follow the instruction in the [Demuxing an MPTS Stream](#) section

#### ➔ To Add a Pull Stream:

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
2. In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.
3. Enter the unique **Stream ID**. for the Broadcaster.



Note: this must be identical (case sensitive) to the stream name configured in the Zixi Feeder.



## Broadcaster Inputs

### Adding Input Streams

4. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.



This setting can be modified without interrupting an existing stream.

5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
6. Select **Pull** for the input stream.

**Add new input stream**

Stream ID:

Max Outputs:

Show in Matrix:

Push  Pull  UDP  Delayed  File  RTMP  RTSP  HLS  TS over HTTP  RIST

**Stream parameters**

Host:  +

Port:

Stream:

Password:

Latency [ms]:

Ignore TLS certificate:

Bind to IP:

Enable time shift

Transmit as multicast

Decrypt stream

Enable encryption

Recording parameters

Ok Cancel


7. Specify the **Stream Parameters** (see Input Stream Parameter table).
8. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameter table). The stream will be recorded and broadcasted after the specified delay. The time shift feature is typically used to broadcast a live event across time zones (due to the time differences between locations).
9. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameter table).
10. If you want to configure the **Recording Parameters**, specify them in the respective fields (see Input Stream Parameter table).
11. You can decrypt an encrypted stream as it enters the Zixi Broadcaster. To decrypt a stream, select the **Decrypt Stream** checkbox and fill in the parameters as describes in the Input Stream Parameter Table below. For more information, see [Decrypting an Input Stream in Zixi Broadcaster](#).

## Broadcaster Inputs

### Adding Input Streams

12. Click **OK**. The “Input added” message appears on the top of the screen. The Push Stream from the Zixi Feeder is added to the Zixi Broadcaster and now appears in the list of streams available in Zixi Broadcaster inputs UI.

### Input Stream Parameters table

Parameter	Description
<b>Stream Parameters</b> The general stream settings.	
<b>Host</b>	Enter the IP Address of the remote Broadcaster server. Note that additional (failover) destinations can be added by clicking the adjacent “+” button
<b>Port</b>	Enter the port through which the Broadcaster receives the stream from another Broadcaster server. Default: 2088 The default input port for Zixi Broadcaster is UDP port <b>2088</b> . This port <i>must be open</i> on any firewalls between Zixi Broadcaster and other devices it is communicating with.
<b>Stream</b>	Enter the unique stream ID from another Broadcaster server.
<b>Password (Optional)</b>	If desired, enter the string for authentication. It must be identical to the string configured on the other Zixi Broadcaster.
<b>Latency [ms]</b>	Specify the maximum number of milliseconds that Zixi Broadcaster should protect the stream for. For example, 6000 milliseconds would ensure that the stream is protected for up to six seconds of delay in the network. This setting affects memory usage as higher levels of protection require more buffering. Default:6000.
<b>Ignore TLS certificate error</b>	Select this option when connecting to a trusted server that does not have a valid certificate. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors.  This setting can be modified without interrupting an existing stream.
<b>Bind to IP</b>	Select from the drop-down list the local IP address to be used for this Input, OR select <i>Any</i> to enable any IP to be used.
<b>Bind to NIC</b>	Select from the drop-down list the NIC to be used for this Input, OR select <i>Any</i> to enable any NIC to be used.
<b>Enable time shift</b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).

## Broadcaster Inputs

---

### Adding Input Streams

---

---

**Maximum delay [hh:mm:ss] (for Time Shift enabled)** Specify the time of a delayed transmission of this stream. Maximum: 24 hours

---

**Transmit as multicast** Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi Broadcaster that will pull the stream will receive it in multicast. By default Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling **Multicast Pull** in the **Settings**.



This setting can be modified without interrupting an existing stream.

---

**Multicast-Only (for multicast enabled)** If desired, specify multicast-only outputs.

---

**Decrypt stream** Select this checkbox to decrypt the Input stream. Please note that the stream will not be encrypted as it passes through the Zixi path. For more information, see [Decrypting an Input Stream in Zixi Broadcaster](#).

---

Specify the type of encryption that is being used in the stream.

**Decryption type (for Decrypt stream enabled)**



This setting can be modified without interrupting an existing stream.

---

Specify the key for decryption.

**Decryption key (for Decrypt stream enabled)**



This setting can be modified without interrupting an existing stream.

---

**Enable Encryption** Select this checkbox to encrypt the Input stream. For more information, see [Encrypting an Input Stream in Zixi Broadcaster](#).

---

**Encryption type (for Encryption enabled)** Specify the type of Encryption (AES 128/192/256).

---

**Encryption key (for Encryption enabled)** Click **Generate** to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.

---

## Broadcaster Inputs

---

### Adding Input Streams

<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.
<b>Destination Type</b>	<p>Select the radio button for the desired storage type and then fill in the relevant parameters.</p> <ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see <a href="#">Using an External Storage</a> section.</li><li>• <b>Record to S3</b> – the file is saved to your AWS S3 Bucket.</li></ul>
<b>Record to Disk Parameters</b>	
<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours
<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.
<b>Record to S3 Parameters</b>	

## Broadcaster Inputs

---

### Adding Input Streams

---

<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>URL</b>	Specify the URL of the S3 bucket.
<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.

## Adding UDP Streams

This enables the Broadcaster to receive a UDP stream. The UDP stream is not protected by the Zixi protocol. The UDP streams can be one of the following types:

- **UDP** – the input stream is plain UDP.
- **RTP** – the input stream uses the RTP protocol, which adds RTP headers to the UDP stream. RTP adds a 12-byte header, which includes a sequence number and a timestamp that can be used to detect dropped packets. The sequence number in the RTP headers can be used to do automatic reordering and the timestamps can help overcome inter-packet gaps, also known as IAT (Inter Arrival Time). However, plain RTP does not feature error recovery or error protection. By default, the RTP headers (which include the timestamps and sequences) are being used on the input side, but then they are stripped off and they no longer propagate to outputs.
- **RTP + SMPTE 2022 FEC** – the input stream includes forward error correction, which enables error recovery or error protection.

If the stream is an MPTS stream and you want to demux it into separate SPTS input streams, create the stream, by following the instructions below, and then follow the instruction in the [Demuxing an MPTS Stream](#) section.

### ➔ To Add a UDP Stream:

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
2. In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.
3. Enter the unique **Stream ID**.



Note this must be identical (case sensitive) to the stream name configured in the Zixi Feeder.

4. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.

## Broadcaster Inputs

### Adding Input Streams

6. Select **UDP** for the input stream.

The screenshot shows a dialog box titled "Add new input stream". At the top, there are fields for "Stream ID", "Max Outputs" (set to "Unlimited"), and "Show in Matrix" (checked). Below these are radio buttons for different input types: "Push", "Pull", "UDP" (selected), "Delayed", "File", "RTMP", "RTSP", "HLS", "TS over HTTP", and "RIST". The "Stream parameters" section contains several fields: "Port", "Multicast IP", "SSM source IP", "Type" (a dropdown menu set to "UDP"), "Bind to IP" (a dropdown menu set to "Any"), and "Max Bitrate [kbps]" (set to "8000"). Below the parameters are four checkboxes: "Enable time shift", "Transmit as multicast", "Enable encryption", and "Recording parameters", all of which are currently unchecked. At the bottom right of the dialog are "Ok" and "Cancel" buttons.

7. Specify the **Stream Parameters** (see Input Stream Parameters Table).
8. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameters Table).
9. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameters Table).
10. If you want to configure the **Recording Parameters**, specify them in the respective fields (see Input Stream Parameters Table).
11. If you want to **Enable encryption**, select this checkbox (see Input Stream Parameters Table). For more information, see [Encrypting an Input Stream in Zixi Broadcaster](#).
12. Click **OK**.

The "Input added" message appears on the top of the screen. The UDP Stream from the Zixi Feeder is added to the Zixi Broadcaster and now appears in the list of streams available in the Zixi Broadcaster inputs UI.

### Input Stream Parameters Table

Parameter	Description
<b>Stream Parameters</b>	
<b>Port</b>	Specify the port to listen on. This port must be open on all firewalls between the Zixi Broadcaster and other devices it is communicating with.
<b>Multicast IP</b>	If multicast, type the Multicast IP address (or leave blank in the case of Unicast).

## Broadcaster Inputs

### Adding Input Streams

<b>SSM Source</b>	Specify the source IP for a source-specific multicast.
<b>Type</b>	<p>Select the appropriate protocol type from the drop-down list:</p> <p><b>UDP</b> – the input stream is plain UDP.</p> <p><b>RTP</b> – the input stream uses the RTP protocol, which adds RTP headers to the UDP stream. RTP adds a 12-byte header, which includes a sequence number and a timestamp that can be used to detect dropped packets. The sequence number in the RTP headers can be used to do automatic reordering and the timestamps can help overcome inter-packet gaps, also known as IAT (Inter Arrival Time). However, plain RTP does not feature error recovery or error protection. By default, the RTP headers (which include the timestamps and sequences) are being used on the input side, but then they are stripped off and they no longer propagate to outputs.</p> <p><b>RTP + SMPTE 2022 FEC</b> – the input stream includes forward error correction, which enables error recovery or error protection.</p>
<b>Bind to IP</b>	Select from the drop-down list the local IP address to be used for this Input, OR select <i>Any</i> to enable any IP to be used.
<b>Max. bitrate [kbps]</b>	<p>Specify the maximum expected bitrate for memory allocation. Recommended: 2X the actual bitrate, which will prevent buffer overruns (especially with VBR streams). Default: 8000.</p> <p>Note – Overflows will typically occur when the Max Bitrate isn't sufficient.</p>
<b>Enable time shift</b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).
<b>Maximum delay [hh:mm:ss] (for Time Shift enabled)</b>	Specify the maximum time delay of this input.
<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling Multicast Pull in the Settings.
<b>Multicast-Only (for multicast enabled)</b>	Selecting this checkbox will force the transmission of this stream only in multicast.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a>



## Broadcaster Inputs

---

### Adding Input Streams

---

---

**Encryption type  
(for Encryption  
enabled)**

Specify the type of Encryption (AES 128/192/256).

---

**Encryption key  
(for Encryption  
enabled)**

Click **Generate** to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.

---

**Recording  
parameters**

Select this checkbox to customize the recording parameters.

---

**Destination Type**

Select the radio button for the desired storage type and then fill in the relevant parameters.

- **Record to disk** – the recording is saved on the local disk. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see [Using an External Storage](#) section.
  - **Record to S3** – the file is saved to your AWS S3 Bucket.
- 

**Record to Disk  
Parameters**

---

**File Name  
Template**

Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.

The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:

- %S=stream id - optional.
- %Y=year - optional
- %M=month - optional
- %D=day - optional
- %T=HH.MM.SS creation time - mandatory

For example, the following pattern %Y\_recording%T.ts will generate the following file name 2017\_recording15:32:35.ts.

---

**Max recorded file  
duration [hours]**

Specify the maximum time allotted for recording a video stream. Default: 2 hours

---

**Keep recorded  
files for [hours]**

Specify the maximum time for storing the recording. Default: 0 hours

---

## Broadcaster Inputs

---

### Adding Input Streams

---

<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.
<b>Record to S3 Parameters</b>	
<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>URL</b>	Specify the URL of the S3 bucket.
<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.

## Adding Delayed Streams

You can delay the broadcasting of an input stream by selecting **Enable time shift** and then creating a Delayed input stream that uses the Input with the time shift as a source. In this case the input stream will be stored on the local disk, while a variant of the input is created and delayed according to the specified duration. The input stream of the time-shifted stream will be stored on the local disk. In parallel, you can create a "Delayed" stream that will stream the "time-shifted" input stream according to the specified delay. For example, you can specify a 6-hour time-shift to "stream\_A". This means that 6 hours of streaming are stored. You can then create a delayed stream that will stream "Stream\_A" with a two-hour delay. This means that the delayed stream will always stream with a two-hour delay.



The duration of the delayed stream should be less or equal to the time shifted stream.

### ➔ To Add a Delayed Stream:

1. Make sure at least one of the existing inputs has the **Enable time shift** setting selected.
2. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
3. In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.
4. Enter the unique **Stream ID**.
5. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
6. Select the **Show in Matrix** check box to display this stream in the Matrix (see Using the Matrix). Default: Selected.
7. Select **Delayed** for the input stream.

**Add new input stream** [X]

Stream ID:

Max Outputs:

Show in Matrix:

Push  Pull  UDP  Delayed  File  RTMP  RTSP  HLS  TS over HTTP

**Stream parameters**

Source:

Delay [hh:mm:ss]:

Enable time shift

Transmit as multicast

Ok Cancel

8. Specify the **Stream Parameters** (see Input Stream Parameters Table).
9. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameters Table).

## Broadcaster Inputs

---

### Adding Input Streams

10. Click **OK**.

The “Input added” message appears on the top of the screen. The Delayed Stream is added to the Zixi Broadcaster and now appears in the list of streams available in the Zixi Broadcaster inputs UI.

### Input Stream Parameters Table

Parameter	Description
<b>Stream Parameters</b>	
<b>Source</b>	Select one of the input streams that were configured to enable "time shift".
<b>Delay</b>	Specify the delay period relative to the original stream in the following format: hh:mm:ss.
<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream’s outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling <b>Multicast Pull</b> in the <b>Settings</b> .
<b>Multicast-Only (for Multicast enabled)</b>	If desired, specify multicast only outputs.


## Adding File Streams

This input creates a file input for the Zixi Broadcaster from a local or mounted disk and subsequently streams it to the end-user.

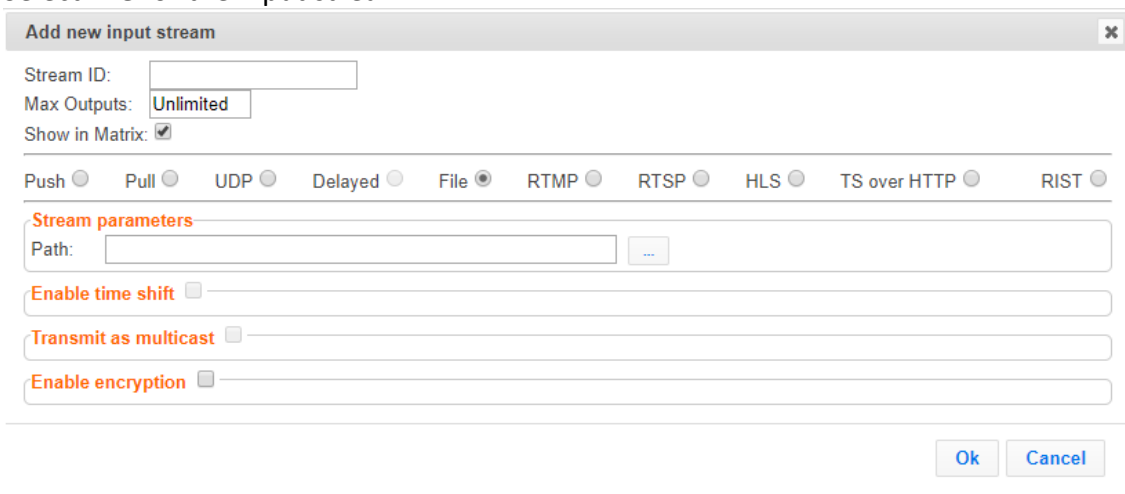
If the stream is an MPTS stream, you can either upload it as an MPTS file or you can demux it into separate SPTS input streams. To Demux the MPTS stream, create the input stream, by following the instructions below, and then follow the instruction in the [Demuxing an MPTS Stream](#) section.

### ➔ To Add a File Stream:

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.  
In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.
2. Enter the unique **Stream ID**.

 Note: this must be identical (case sensitive) to the stream name configured in the Zixi Feeder.

4. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
6. Select **File** for the input stream.



7. Specify the **Stream Parameters** (see Input Stream Parameter table).
8. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameter table). The stream will be recorded and broadcasted after the specified delay. The time shift feature is typically used to broadcast a live event across time zones (due to the time differences between locations).
9. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameter table).

## Broadcaster Inputs

### Adding Input Streams

10. If you want to **Enable encryption**, select this checkbox (see Input Stream Parameter table). For more information, see [Encrypting an Input Stream in Zixi Broadcaster](#).
11. Click **OK**. The “Input added” message appears on the top of the screen. The Push Stream from the Zixi Feeder is added to the Zixi Broadcaster and now appears in the list of streams available in the Zixi Broadcaster inputs UI.

### Input Stream Parameters Table

Parameter	Description
<b>Stream Parameters</b>	
<b>Path</b>	Type the local or mounted (Linux) disk or click the adjacent browse button to select the desired one. The suffix .ts indicates a transmitted stream.
<b>Enable time shift</b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).
<b>Maximum delay [hh:mm:ss] (for Time Shift enabled)</b> Specify the maximum time delay of this input.	
<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default, Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling Multicast Pull in the Settings.
<b>Multicast Only (for Multicast enabled)</b>	If desired, specify force multicast-only outputs. A Zixi receiver will receive it in multicast (unless it is out of the LAN in which case it will relay in as unicast). You can also force Multicast only outputs.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a> .
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).
<b>Encryption key (for encryption enabled)</b>	Click <b>Generate</b> to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.

## Broadcaster Inputs

### Adding Input Streams

## Adding RTMP Streams

This enables you to create a Push or Pull RTMP input stream.

### ➔ To Add a RTMP Stream:

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
2. In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.
3. Enter the unique **Stream ID**.



Note: this must be identical (case sensitive) to the stream name configured in the Zixi Feeder.

4. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
6. Select **RTMP** for the input stream.

**Add new input stream** [Close]

Stream ID:

Max Outputs:

Show in Matrix:

Push  Pull  UDP  Delayed  File  RTMP  RTSP  HLS  TS over HTTP  RIST

**Stream parameters**

Type: Push  Pull

Disconnect if inactive:

Padding:

Muxing Bitrate [kbps]:

Smoothing Latency [ms]:

Enable time shift

Transmit as multicast

Enable encryption

Recording parameters

Ok Cancel

7. Specify the **Stream Parameters** (see Input Stream Parameters Table).
8. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameters Table).
9. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameters Table).
10. If you want to **Enable encryption**, select this checkbox (see Input Stream Parameters Table). For more information, see [Encrypting an Input Stream in Zixi Broadcaster](#).

## Broadcaster Inputs

---

### Adding Input Streams

11. If you want to configure the **Recording Parameters**, specify them in the respective fields (see Input Stream Parameters Table).
12. Click **OK**. The “Input added” message appears on the top of the screen. The RTMP Stream is added to the Zixi Broadcaster and now appears in the list of streams available in the Zixi Broadcaster inputs UI.

### Input Stream Parameters Table

Parameter	Description
<b>Stream Parameters</b>	
<b>Type</b>	Select the desired type for the RTMP stream: <ul style="list-style-type: none"><li>• Push – the stream is pushed to the Zixi Broadcaster (not initiated by the Zixi Broadcaster). No additional connection parameters need to be filled. If the stream name will match, the stream will be inputted.</li><li>• Pull – Zixi Broadcaster will initiate the stream by accessing it through the following parameters:</li></ul>
<b>URL (for Pull)</b>	Type the Stream URL from which the Zixi Broadcaster will access. Use the following URL format: <code>rtmp://host:[port]/app/stream</code> .  You can define RTMPS clients with the proven security of a secure socket layer (SSL), by using the <code>rtmps:// url</code>
<b>Stream Name (for Pull)</b>	Type the unique name for the stream that has been defined in the RTMP server.
<b>User Name (for Pull)</b>	Specify the user name to authenticate on the remote RTMP server
<b>Password (for Pull)</b>	Specify the string that is used for authorization on the remote RTMP server.
<b>Ignore TLS certificate</b>	Select this option when connecting to a trusted server that does not have a valid certificate. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors.
<b>Disconnect if inactive</b>	Disconnects the connection if inactive.
<b>Padding</b>	If the stream's bitrate fluctuates you can pad the stream with null packets to create a steady bitrate. Select this checkbox to enable padding.



## Broadcaster Inputs

### Adding Input Streams

<b>Muxing bitrate (for Padding enabled)</b>	Specify the padding level in kbps.
<b>Smoothing Latency [ms]</b>	The Smoothing feature uses a buffer to time packets according to the stream's clock. This is useful when there are bursts in the rate in which the stream arrives. The smoothing features creates a queue in the buffer, allowing smoother entry into the Broadcaster. The Smoothing parameter defines the size of the buffer in Milliseconds. A larger buffer enables smoother entry, however, it may influence the latency rate.
<b>Enable time shift</b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).
<b>Maximum delay [hh:mm:ss] (for Time Shift enabled)</b>	Specify the maximum time delay of this input.
<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default, Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling Multicast Pull in the Settings.
<b>Multicast Only (for Multicast enabled)</b>	Select the check box to force only Multicast outputs.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a>
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).
<b>Encryption key (for Encryption enabled)</b>	Click <b>Generate</b> to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.
<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.
	Select the radio button for the desired storage type and then fill in the relevant parameters.
<b>Destination Type</b>	<ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see <a href="#">Using an External Storage</a> section.</li></ul>

## Broadcaster Inputs

---

### Adding Input Streams

---

- **Record to S3** – the file is saved to your AWS S3 Bucket.
- 

#### **Record to Disk Parameters**

---

<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
---------------------------	---

---

<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
---	--

---

<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours
--	--

---

<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.
--------------------	--

---

#### **Record to S3 Parameters**

---

<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li></ul>
---------------------------	--

---

## Broadcaster Inputs

---

### Adding Input Streams

---

- %M=month - optional
- %D=day - optional
- %T=HH.MM.SS creation time - mandatory

For example, the following pattern %Y\_recording%T.ts will generate the following file name 2017\_recording15:32:35.ts.

---

<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
---	--

---

<b>URL</b>	Specify the URL of the S3 bucket.
------------	-----------------------------------

---

<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
--------------------------------------	---

---

<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
-------------------	--

---

<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.
-------------------	--

---

## Broadcaster Inputs

### Adding Input Streams

## Adding RTSP Streams


This enables you to create RTSP streams.

### ➔ To Add RTSP Streams:

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
2. In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.

The screenshot shows the 'Add new input stream' dialog box. The 'Stream ID' field is empty. The 'Max Outputs' dropdown is set to 'Unlimited'. The 'Show in Matrix' checkbox is checked. The 'RTSP' radio button is selected. The 'Stream parameters' section shows the URL field is empty, 'TCP' is checked, and the UDP ports are 5000 for video and 5002 for audio. The 'Enable time shift', 'Transmit as multicast', 'Enable encryption', and 'Recording parameters' checkboxes are all unchecked.

3. Enter the unique **Stream ID**.

4.  Note: this must be identical (case sensitive) to the stream name configured in the Zixi Feeder.

5. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
6. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
7. Select **RTMP** for the input stream.
8. Specify the **Stream Parameters** (see Input Stream Parameters Table).
9. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameters Table).
10. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameters Table).
11. If you want to **Enable encryption**, select this checkbox (see Input Stream Parameters Table).
12. If you want to configure the **Recording Parameters**, specify them in the respective fields (see Input Stream Parameters Table).

## Broadcaster Inputs

---

### Adding Input Streams

13. Click **OK**. The “Input added” message appears on the top of the screen. The RTMP Stream is added to the Zixi Broadcaster and now appears in the list of streams available in the Zixi Broadcaster inputs UI.

### Input Stream Parameters Table

Parameter	Description
<b><i>Stream Parameters</i></b>	The general stream settings.
<b>URL</b>	Type the Stream URL from which the Zixi Broadcaster will access.
<b>Video UDP Port</b>	Specify the UDP port to be used for streaming the video.
<b>Audio UDP Port</b>	Specify the UDP port to be used for streaming the audio.
<b>Pad to bitrate</b>	If the stream's bitrate fluctuates you can pad the stream with null packets to create a steady bitrate. This setting determines the padding level in kbps.
<b>Latency</b>	Specify the maximum latency you can tolerate for the stream in milliseconds. Recommended - the length of two frames.
<b>TCP</b>	Select this checkbox for streams that use TCP protocol. Leave unselected in case the stream uses UDP.
<b><i>Enable time shift</i></b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).
<b>Maximum delay [hh:mm:ss] (for Time Shift enabled)</b>	Specify the maximum time delay of this input.
<b><i>Transmit as multicast</i></b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling Multicast Pull in the Settings.
<b>Multicast Only (for Multicast enabled)</b>	Select the check box to force only Multicast outputs.
<b><i>Enable Encryption</i></b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a> .
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).

## Broadcaster Inputs

---

### Adding Input Streams

---

<b>Encryption key (for Encryption enabled)</b>	Click <b>Generate</b> to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.
<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.
<b>Destination Type</b>	<p>Select the radio button for the desired storage type and then fill in the relevant parameters.</p> <ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see <a href="#">Using an External Storage</a> section.</li><li>• <b>Record to S3</b> – the file is saved to your AWS S3 Bucket.</li></ul>
<b>Record to Disk Parameters</b>	
<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours
<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.

## Broadcaster Inputs

---

### Adding Input Streams

---

---

#### **Record to S3 Parameters**

---

---

<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>URL</b>	Specify the URL of the S3 bucket.
<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.

---

---

## Broadcaster Inputs

### Adding Input Streams

## Adding HLS Streams

This screen enables the Broadcaster to receive an HLS stream. The HLS stream is not protected by the Zixi protocol. There are two options for pulling the stream:

- **Single source (adaptive bitrate)** - the stream can be an adaptive bitrate stream. In this case, Zixi Broadcaster will pull the main playlist to adapt between bitrates according to network conditions.
- **Multiple sources** - it is possible to pull multiple input streams with static individual bitrates. In this case, Zixi Broadcaster will automatically create a separate input for each bitrate.

### ➔ To Add an HLS stream:

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
2. In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.
3. Enter the unique **Stream ID**.



Note: this must be identical (case sensitive) to the stream name configured in the Zixi Feeder.

4. Enter the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
6. Select **HLS** for the input stream.

The screenshot shows the 'Add new input stream' dialog box. The 'Stream ID' field is empty. The 'Max Outputs' dropdown is set to 'Unlimited'. The 'Show in Matrix' checkbox is checked. The 'HLS' radio button is selected. The 'Master playlist URL' is 'http://localhost:7777/input.m3u8'. The 'Download type' is 'Multiple sources (pull all bitrates)'. The 'Smoothing Latency' is '30000'. There are checkboxes for 'Inject SCTE35 on CUE marks', 'Enable time shift', 'Transmit as multicast', 'Enable encryption', and 'Recording parameters'. The 'Ok' and 'Cancel' buttons are at the bottom right.

7. Specify the **Stream Parameters** (see Input Stream Parameters Table below).



## Broadcaster Inputs

### Adding Input Streams

8. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameters Table).
9. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameters Table).
10. If you want to configure the **Recording Parameters**, specify them in the respective fields (see Input Stream Parameters Table).
11. If you want to **Enable encryption**, select this checkbox (see Input Stream Parameters Table).
12. Click **OK**.

The "Input added" message appears on the top of the screen. The HLS stream from the Zixi Feeder is added to the Zixi Broadcaster and now appears in the list of streams available in the Zixi Broadcaster inputs UI. If the "multiple sources" option was select, Zixi Broadcaster will automatically create a separate input for each bitrate.

### Input Stream Parameters Table

Parameter	Description
<b>Stream Parameters</b>	
<b>Master Playlist URL</b>	Specify the URL of the HLS stream.
<b>Download Type</b>	Specify whether the variants will be pulled from a single source (adaptive bitrate) or multiple sources (pull all bitrates).
<b>Smoothing Latency</b>	The Smoothing feature uses a buffer to time packets according to the stream's clock. This is useful when there are bursts in the rate in which the stream arrives. The smoothing features creates a queue in the buffer, allowing smoother entry into the Broadcaster. The Smoothing parameter defines the size of the buffer in Milliseconds. A larger buffer enables smoother entry, however, it may influence the latency rate.
<b>Inject SCTE35 on CUE Marks</b>	Select this checkbox to inject SCTE markers into the adaptive stream manifest. SCTE-35 markers found in the source input will be used to segment the stream and the #EXT-X-CUE-IN/#EXT-X-CUE-OUT tags will be added to the manifest.
<b>Enable time shift</b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).
<b>Maximum delay [hh:mm:ss] (for</b>	Specify the maximum time delay of this input.

## Broadcaster Inputs

---

### Adding Input Streams

---

<b>Time Shift enabled)</b>	
<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default, Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling Multicast Pull in the Settings.
<b>Multicast Only (for Multicast enabled)</b>	If desired, specify multicast only outputs.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a> .
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).
<b>Encryption key (for Encryption enabled)</b>	Click <b>Generate</b> to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.
<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.
<b>Destination Type</b>	Select the radio button for the desired storage type and then fill in the relevant parameters. <ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see <a href="#">Using an External Storage</a> section.</li><li>• <b>Record to S3</b> – the file is saved to your AWS S3 Bucket.</li></ul>
<b>Record to Disk Parameters</b>	
<b>File Name Template</b>	Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.  The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows: <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li></ul>

## Broadcaster Inputs

---

### Adding Input Streams

---

- %M=month - optional
- %D=day - optional
- %T=HH.MM.SS creation time - mandatory

For example, the following pattern %Y\_recording%T.ts will generate the following file name 2017\_recording15:32:35.ts.

---

**Max recorded file duration [hours]** Specify the maximum time allotted for recording a video stream. Default: 2 hours

---

**Keep recorded files for [hours]** Specify the maximum time for storing the recording. Default: 0 hours

---

**Custom Path** Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the **Settings > General** screen.

---

### **Record to S3 Parameters**

---

Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.

The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:

#### **File Name Template**

- %S=stream id - optional.
- %Y=year - optional
- %M=month - optional
- %D=day - optional
- %T=HH.MM.SS creation time - mandatory

For example, the following pattern %Y\_recording%T.ts will generate the following file name 2017\_recording15:32:35.ts.

---

**Max recorded file duration [hours]** Specify the maximum time allotted for recording a video stream. Default: 2 hours

---

**URL** Specify the URL of the S3 bucket.

---

**Ignore TLS certificate errors** TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.

---

## Broadcaster Inputs

### Adding Input Streams

<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.

## Adding TS over HTTP (Pull) Streams

In this mode Zixi Broadcaster pulls a Transport Stream over TCP from an HTTP server. In this case, the stream is initiated by your Zixi Broadcaster.

### ➔ To Add a TS over HTTP (Pull) Stream:

1. In the Zixi Broadcaster navigation menu, click the **Inputs** tab.
2. In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.
3. Enter the unique **Stream ID**.



Note: this must be identical (case sensitive) to the stream name configured in the Zixi Feeder.

1. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
2. Select the **Show in Matrix** checkbox to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
3. Select **TS over HTTP** for the input stream.
4. In the **Type** field, select **Pull**.

**Add new input stream**

Stream ID:

Max Outputs:

Show in Matrix:

Push  Pull  UDP  Delayed  File  RTMP  RTSP  HLS  TS over HTTP  RIST

**Stream parameters**

Type:  Push  Pull

URL (http://host:[port]/stream):

Smoothing [ms]:

Enable time shift

Transmit as multicast

Enable encryption

Recording parameters

5. Specify the **Stream Parameters** (see Input Stream Parameter table).

## Broadcaster Inputs

---

### Adding Input Streams

6. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameter table). The stream will be recorded and broadcasted after the specified delay. The time shift feature is typically used to broadcast a live event across time zones (due to the time differences between locations).
7. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameter table).
8. If you want to **Enable encryption**, select this checkbox (see Input Stream Parameters Table).
9. If you want to configure the **Recording Parameters**, specify them in the respective fields (see Input Stream Parameter table).
10. Click **OK**.

### Input Stream Parameters table

Parameter	Description
<b>Stream Parameters</b>	The general stream settings.
<b>TS URL</b>	Enter the URL of the Transport Stream.
<b>Smoothing</b>	The Smoothing feature uses a buffer to time packets according to the stream's clock. This is useful when there are bursts in the rate in which the stream arrives. The smoothing features creates a queue in the buffer, allowing smoother entry into the Broadcaster. The Smoothing parameter defines the size of the buffer in Milliseconds. A larger the buffer enables smoother entry. However, it may influence the latency rate.
<b>Enable time shift</b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).
<b>Maximum delay [hh:mm:ss] (for Time Shift enabled)</b>	Specify the time of a delayed transmission of this stream. Maximum: 24 hours
<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi Broadcaster that will pull the stream will receive it in multicast. By default, Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling <b>Multicast Pull</b> in the <b>Settings</b> .
<b>Multicast-Only (for multicast enabled)</b>	If desired, specify multicast-only outputs.

## Broadcaster Inputs

---

### Adding Input Streams

<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a> .
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).
<b>Encryption key (for Encryption enabled)</b>	Click Generate to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.
<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.
<b>Destination Type</b>	<p>Select the radio button for the desired storage type and then fill in the relevant parameters.</p> <ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see <a href="#">Using an External Storage</a> section.</li><li>• <b>Record to S3</b> – the file is saved to your AWS S3 Bucket.</li></ul>
<b>Record to Disk Parameters</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>File Name Template</b>	
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours

## Broadcaster Inputs

---

### Adding Input Streams

---

<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours
<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.
<b>Record to S3 Parameters</b>	
<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>URL</b>	Specify the URL of the S3 bucket.
<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.

---

## Adding TS over HTTP (Push) Streams

In this mode Zixi Broadcaster will receive a Transport Stream over TCP from an HTTP server. In this case, the stream is initiated by the source HTTP Server, while the Zixi Broadcaster remains in “listening mode” until the stream (with the specified stream ID) is initiated.

## Broadcaster Inputs

### Adding Input Streams

#### ➔ To Add a TS over HTTP (Push) Stream:

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
2. In the **Menu** bar, click **New Input**. The **Add a new input stream** window is displayed.
3. Enter the unique **Stream ID** for the Broadcaster.



Note: this must be identical (case sensitive) to the stream name configured in the Zixi Feeder.

4. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
5. Select the **Show in Matrix** checkbox to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
6. Select **TS over HTTP** for the input stream
7. In the **Type** field, select **Push**.

8. Specify the **Stream Parameters** (see Input Stream Parameters Table below).
9. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameter Table below). The stream will be recorded and broadcasted after the specified delay. The time shift feature is typically used to broadcast a live event across time zones (due to the time differences between locations).
10. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameter Table).
11. If you want to **Enable encryption**, select this checkbox (see Input Stream Parameters Table).
12. If you want to record the stream, you can configure the **Recording Parameters** (see Input Stream Parameter Table).
13. Click **OK**.



## Broadcaster Inputs

### Adding Input Streams

#### Input Stream Parameters Table

Parameter	Description
<b>Stream Parameters</b>	
<b>Smoothing</b>	The Smoothing feature uses a buffer to time packets according to the stream's clock. This is useful when there are bursts in the rate in which the stream arrives. The smoothing features creates a queue in the buffer, allowing smoother entry into the Broadcaster. The Smoothing parameter defines the size of the buffer in Milliseconds. A larger buffer enables smoother entry, however, it may influence the latency rate.
<b>User</b>	If the stream is password protected, enter the username to access the stream.
<b>Password</b>	If the stream is password protected, enter the password.
<b>Enable time shift</b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).
<b>Maximum delay [hh:mm:ss]</b>	Specify the time of the delayed transmission of this stream. Maximum: 24 hours.
<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default, Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling <b>Multicast Pull</b> in the <b>Settings</b> .
<b>Multicast Only</b>	Selecting this checkbox will force the transmission of this stream only in multicast.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a> .
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).
<b>Encryption key (for Encryption enabled)</b>	Click Generate to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.
<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.

## Broadcaster Inputs

---

### Adding Input Streams

---

	Select the radio button for the desired storage type and then fill in the relevant parameters.
<b>Destination Type</b>	<ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see <a href="#">Using an External Storage</a> section.</li><li>• <b>Record to S3</b> – the file is saved to your AWS S3 Bucket.</li></ul>

---

#### **Record to Disk Parameters**

---

	Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.
<b>File Name Template</b>	<p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>

---

<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
---	--

---

<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours
--	--

---

<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.
--------------------	--

---

#### **Record to S3 Parameters**

---

## Broadcaster Inputs

---

### Adding Input Streams

---

<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>URL</b>	Specify the URL of the S3 bucket.
<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.

---

## Adding RIST Streams

This enables the Broadcaster to receive a RIST stream. RIST seeks to provide reliable, high performance media transport by using UDP at the transport layer to avoid the inefficiencies of TCP, then layering on top Forward Error Correction and Retransmits to add the reliability inherently absent from UDP. Zixi Broadcaster supports RIST streams from any compliant RIST device.

Zixi supports the following RIST profiles:

- **Simple Profile** - The RIST Simple Profile provides best-in-class packet protection, as well as support for multi-link bonding and seamless switching.
- **Main Profile** – The RIST Main Profile builds upon the features of the Simple Profile to provide backwards compatibility, while opening the door for additional features, including the use of industry-standard DTLS with a variety of available cyphers to ensure the most up-to-date protection, while being flexible to accommodate local legal regulations. Support for pre-shared keys is also included, which makes RIST Main Profile compatible with one-to-many environments such as satellite or multicast distribution

Contact your Zixi representative to learn more about interoperability testing between Zixi Broadcaster and 3rd party devices. More details about RIST can be found at [www.rist.tv](http://www.rist.tv)

### ➔ To Add a RIST Stream:

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
2. In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.
3. Enter the unique **Stream ID**.



Note this must be identical (case sensitive) to the stream name configured at the source (e.g. Zixi Feeder).

4. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.

## Broadcaster Inputs

### Adding Input Streams

6. Select **RIST** for the input stream.

The screenshot shows the 'Add new input stream' dialog box. At the top, there are fields for 'Stream ID', 'Max Outputs' (set to 'Unlimited'), and 'Show in Matrix' (checked). Below these are radio buttons for different input types: Push, Pull, UDP, Delayed, File, RTMP, RTSP, HLS, TS over HTTP, and RIST (which is selected). The 'Stream parameters' section contains several fields: Profile (Simple Profile), Local Port, Multicast IP, SSM source IP, Remote Port (0=auto), Type (RTP), Bind to IP (Any), Max Bitrate (8000), and Max Latency (1000). There are also four checkboxes: 'Enable time shift', 'Transmit as multicast', 'Enable encryption', and 'Recording parameters'. At the bottom right, there are 'Ok' and 'Cancel' buttons.

7. Specify the **Stream Parameters** (see Input Stream Parameters Table).
8. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameters Table).
9. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameters Table).
10. If you want to configure the **Recording Parameters**, specify them in the respective fields (see Input Stream Parameters Table).
11. If you want to **Enable encryption**, select this checkbox (see Input Stream Parameters Table). For more information, see [Encrypting an Input Stream in Zixi Broadcaster](#).
12. Click **OK**.

The “Input added” message appears on the top of the screen. The RIST Stream is added to the Zixi Broadcaster and now appears in the list of streams available in the Zixi Broadcaster inputs UI.

### Input Stream Parameters Table

Parameter	Description
<b>Stream Parameters</b>	

## Broadcaster Inputs

### Adding Input Streams

<b>Profile</b>	Select one of the following RIST profiles that matches the input stream: <ul style="list-style-type: none"><li>• Simple Profile (default)</li><li>• Main Profile</li></ul>
<b>Simple Profile Parameters</b>	
<b>Local Port</b>	Specify the port to listen on. This port must be open on all firewalls between the Zixi Broadcaster and other devices it is communicating with.
<b>Multicast IP</b>	If multicast, type the Multicast IP address (or leave blank in the case of Unicast).
<b>SSM Source</b>	Specify the source IP for a source-specific multicast.
<b>Remote Port</b>	Specify the remote port in the source RIST device. Default: 0 (any port)
<b>Type</b>	Select one of the following protocol types from the drop-down list: <ul style="list-style-type: none"><li>• RTP</li><li>• RTP + SMPTE 2022 FEC</li></ul>
<b>Bind to IP</b>	Select from the drop-down list the local IP address to be used for this Input, OR select <i>Any</i> to enable any IP to be used.
<b>Max. bitrate [kbps]</b>	Specify the maximum expected bitrate for memory allocation. Recommended: 2X the actual bitrate, which will prevent buffer overruns (especially with VBR streams). Default: 8000. Note – Overflows will typically occur when the Max Bitrate isn't sufficient.
<b>Max Latency [ms]</b>	Specify the maximum latency of the stream in milliseconds. Default: 1000
<b>Main Profile – Push Parameters</b>	
<b>Local Port</b>	Specify the port to listen on. This port must be open on all firewalls between the Zixi Broadcaster and other devices it is communicating with.
<b>Multicast IP</b>	If multicast, type the Multicast IP address (or leave blank in the case of Unicast).
<b>SSM Source</b>	Specify the source IP for a source-specific multicast.
<b>Remote Port</b>	Specify the remote port in the source RIST device. Default: 0 (any port)
<b>Type</b>	Select one of the following protocol types from the drop-down list: <ul style="list-style-type: none"><li>• RTP</li><li>• RTP + SMPTE 2022 FEC</li></ul>
<b>Bind to IP</b>	Select from the drop-down list the local IP address to be used for this Input, OR select <i>Any</i> to enable any IP to be used.

## Broadcaster Inputs

### Adding Input Streams

<b>Max. bitrate [kbps]</b>	Specify the maximum expected bitrate for memory allocation. Recommended: 2X the actual bitrate, which will prevent buffer overruns (especially with VBR streams). Default: 8000. Note – Overflows will typically occur when the Max Bitrate isn't sufficient.
<b>Max Latency [ms]</b>	Specify the maximum latency of the stream in milliseconds. Default: 1000
<b>Use DTLS</b>	Select this checkbox to encrypt the stream dynamically using DTLS.
<b>Main Profile – Pull Parameters</b>	
<b>Remote Host</b>	Specify the IP Address of the remote RIST device from which the stream will be pulled.
<b>Remote Port</b>	Specify the port of the remote RIST device from which the stream will be pulled. Default: 0 (any port)
<b>Local Port</b>	Specify the local port to which the stream will be pulled. Default: 0 (any port)
<b>Type</b>	Select one of the following protocol types from the drop-down list: <ul style="list-style-type: none"><li>• RTP</li><li>• RTP + SMPTE 2022 FEC</li></ul>
<b>Bind to IP</b>	Select from the drop-down list the local IP address to be used for this Input, OR select <i>Any</i> to enable any IP to be used.
<b>Max. bitrate [kbps]</b>	Specify the maximum expected bitrate for memory allocation. Recommended: 2X the actual bitrate, which will prevent buffer overruns (especially with VBR streams). Default: 8000. Note – Overflows will typically occur when the Max Bitrate isn't sufficient.
<b>Max Latency [ms]</b>	Specify the maximum latency of the stream in milliseconds. Default: 1000.
<b>Use DTLS</b>	Select this checkbox to encrypt the stream dynamically using DTLS.
<b>Enable time shift</b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).
<b>Maximum delay [hh:mm:ss] (for Time Shift enabled)</b>	Specify the maximum time delay of this input.
<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling Multicast Pull in the Settings.

## Broadcaster Inputs

---

### Adding Input Streams

---

---

**Multicast-Only (for multicast enabled)** Selecting this checkbox will force the transmission of this stream only in multicast.

---

**Enable Encryption** Select this checkbox to encrypt the Input stream. For more information, see [Encrypting an Input Stream in Zixi Broadcaster](#)

---

**Encryption type (for Encryption enabled)** Specify the type of Encryption (AES 128/192/256).

---

**Encryption key (for Encryption enabled)** Click **Generate** to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.

---

**Recording parameters** Select this checkbox to customize the recording parameters.

---

Select the radio button for the desired storage type and then fill in the relevant parameters.

**Destination Type**

- **Record to disk** – the recording is saved on the local disk. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see [Using an External Storage](#) section.
- **Record to S3** – the file is saved to your AWS S3 Bucket.

---

#### **Record to Disk Parameters**

---

Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.

The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:

**File Name Template**

- %S=stream id - optional.
- %Y=year - optional
- %M=month - optional
- %D=day - optional
- %T=HH.MM.SS creation time - mandatory

For example, the following pattern %Y\_recording%T.ts will generate the following file name 2017\_recording15:32:35.ts.

---

---



## Broadcaster Inputs

---

### Adding Input Streams

---

---

**Max recorded file duration [hours]** Specify the maximum time allotted for recording a video stream. Default: 2 hours

---

---

**Keep recorded files for [hours]** Specify the maximum time for storing the recording. Default: 0 hours

---

---

**Custom Path** Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the **Settings > General** screen.

---

---

#### **Record to S3 Parameters**

---

---

**File Name Template** Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.

The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:

- %S=stream id - optional.
- %Y=year - optional
- %M=month - optional
- %D=day - optional
- %T=HH.MM.SS creation time - mandatory

For example, the following pattern %Y\_recording%T.ts will generate the following file name 2017\_recording15:32:35.ts.

---

---

**Max recorded file duration [hours]** Specify the maximum time allotted for recording a video stream. Default: 2 hours

---

---

**URL** Specify the URL of the S3 bucket.

---

---

**Ignore TLS certificate errors** TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.

---

---

**Access Key** The access key that is used for accessing the S3 bucket.

---

---

**Secret Key** The secret key that is used for accessing the S3 bucket.

---

---

## Broadcaster Inputs

### Adding Input Streams

## Adding NDI Streams

The NDI is pulled into Zixi Broadcaster using the auto-discovery mechanism described in the NDI Stream Discovery section.

### ➔ To Add an NDI Input Stream:



Before adding the NDI Input Stream, you must configure the relevant transcoding profiles – See [Configuring Transcoding Profiles for NDI Sources](#).

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
2. In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.
3. Select **NDI** for the input stream.

The screenshot shows the 'Add new input stream' dialog box. It has a title bar with a close button. Below the title bar, there are input fields for 'Stream ID', 'Max Outputs' (set to 'Unlimited'), and a checked 'Show in Matrix' checkbox. A horizontal row of radio buttons allows selecting the input type: Push, Pull, UDP, Delayed, File, RTMP, RTSP, HLS, TS over HTTP, SRT, SRT Client, RIST, and NDI (selected). The 'Stream parameters' section contains a 'Stream' dropdown menu (set to 'KOBYS-IPHONE'), a 'Remove stream' button, 'Audio profile', 'Audio Reference Level [dB]' (set to 0), 'Transport stream bitrate [kbps]', 'Use Intel hardware acceleration (if available)' (unchecked), and 'Smoothing buffer [ms]' (set to 0). Below this are checkboxes for 'Enable time shift', 'Transmit as multicast', 'Enable encryption', and 'Recording parameters'. At the bottom right, there are 'Ok' and 'Cancel' buttons.

4. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
6. Specify the **Stream Parameters** (see Input Stream Parameter table).
7. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameter table). The stream will be recorded and broadcasted after the specified delay. The time shift feature is typically used to broadcast a live event across time zones (due to the time differences between locations).
8. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameter table).
9. If you want to configure the **Recording Parameters**, specify them in the respective fields (see Input Stream Parameter table).
10. If you want to **Enable encryption**, select this checkbox (see Input Stream Parameters Table). For more information, see [Encrypting an Input Stream in Zixi Broadcaster](#).

## Broadcaster Inputs

### Adding Input Streams

11. Click **OK**. The “Input added” message appears on the top of the screen. The NDI Stream is added to the Zixi Broadcaster and now appears in the list of streams available in Zixi Broadcaster Inputs screen.

#### Input Stream Parameters table

Parameter	Description
<b>Stream Parameters</b>	
<b>Stream</b>	Select one of the NDI streams available based on your configuration from the drop-down menu. On a local area network NDI will use the discovery mechanism to advertise the sources and Zixi Broadcaster will automatically discover these streams. To make the streams discoverable in the cloud, you need to deploy an NDI Discovery Server and configure the NDI plug-in with the settings mentioned above.
<b>Video Profile</b>	Select an existing transcoding video profile, or select <b>Remove stream</b> if you want to remove the video track.
<b>Audio Profile</b>	Select an existing transcoding audio profile, or select <b>Remove stream</b> if you want to remove the audio track.
<b>Audio Reference Level [dB]</b>	This parameter attenuates the audio level by the specified level in dB. This setting is used to prevent audio clipping.
<b>Transport Stream Bitrate [kbps]</b>	Specify the bitrate of the transport stream. This parameter is optional. If you do not specify a TS Bitrate, the stream will be VBR. If you specify a bitrate, the stream will be CBR. The bitrate value must be compatible with the selected profile.
<b>Use Intel hardware acceleration (if available)</b>	Select this checkbox to use the Intel hardware acceleration component if it available.
<b>Smoothing buffer</b>	The Smoothing feature uses a buffer to time packets according to the stream's clock. This is useful when there are bursts in the rate in which the stream arrives. The smoothing features creates a queue in the buffer, allowing smoother entry into the Broadcaster. The Smoothing parameter defines the size of the buffer in Milliseconds. A larger the buffer enables smoother entry. However, it may influence the latency rate.
<b>Enable time shift</b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).

## Broadcaster Inputs

---

### Adding Input Streams

---

---

<b>Maximum delay [hh:mm:ss] (for Time Shift enabled)</b>	Specify the time of a delayed transmission of this stream. Maximum: 24 hours
--	--

---

<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi Broadcaster that will pull the stream will receive it in multicast. By default Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling <b>Multicast Pull</b> in the <b>Settings</b> .
------------------------------	--



This setting can be modified without interrupting an existing stream.

---

<b>Multicast-Only (for multicast enabled)</b>	If desired, specify multicast-only outputs.
---	---

---

<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a> .
--------------------------	--

---

<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).
---	---

---

<b>Encryption key (for Encryption enabled)</b>	Click <b>Generate</b> to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.
--	---

---

<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.
-----------------------------	---

---

<b>Destination Type</b>	Select the radio button for the desired storage type and then fill in the relevant parameters. <ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see <a href="#">Using an External Storage</a> section.</li><li>• <b>Record to S3</b> – the file is saved to your AWS S3 Bucket.</li></ul>
-------------------------	---

---

<b>Record to Disk Parameters</b>	
----------------------------------	--

---

## Broadcaster Inputs

---

### Adding Input Streams

---

<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours
<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.
<b>Record to S3 Parameters</b>	

## Broadcaster Inputs

---

### Adding Input Streams

---

<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>URL</b>	Specify the URL of the S3 bucket.
<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.

---

## Broadcaster Inputs

### Viewing Existing Input Streams

## Viewing Existing Input Streams

The **Inputs** screen displays all the input streams that have been created. At the bottom of the screen you can also view accumulative utilization statistics for any connected streams selected. Hovering the cursor on the status indicator of a stream will provide you with the stream configuration information and basic utilization statistics, where applicable.

**Zixi Broadcaster** Version 14.0.38190 Linux 64-bit  
QA-Gil-Bx4x1-2-Setup-Alon-Te-qgle  
Copyright (c) 2007-2020, Zixi LLC.  
CPU: 63.23% Disk: 35.16% Mem: 21.29% Clock: 11:15:09

**STATUS** **INPUTS** **OUTPUTS** **ADAPTIVE** **VOD** **FILES** **TRANSCODER** **SETTINGS** **EVENT LOG** **MATRIX** **LOGOUT**

**Inputs** Search Inputs [x] [P] [G] + New Input Reload

Status	ID	Type	Source	Bitrate[kbps]	Up Time	TR 101 290	Error	Outputs	Actions
Connected	bbb	File	1080p60.ts	0	10:42:22	P1: P2:	None	5	[G] [X]
<b>Transcoded sources</b>									
Status	ID	Type	Source	Bitrate[kbps]	Up Time	TR 101 290	Error	Outputs	Actions
Connected	1			0	04:20:13	Off	None	0	[G] [X]
Offline	bbb-alon-qgle-to-pr0...	Pull	zixi.stagingio.devcloud.zixi.com:2088	0	00:00:00	P1: P2:	Timeout	0	[R] [X]
Connected	bbb-source-alon-tech...	Push	3.16.111.197	7952	00:04:24	P1: P2:	None	0	[G] [X]

**Statistics**

Bitrate [kbps] <a href="#">[L]</a>	Total Packets <a href="#">[L]</a>	FEC Packets <a href="#">[L]</a>
Last Connected: <a href="#">[L]</a>	Packet Rate <a href="#">[L]</a>	FEC Recovered <a href="#">[L]</a>
Reconnections <a href="#">[L]</a>	Packet Loss % <a href="#">[L]</a>	ARQ Requests <a href="#">[L]</a>
RTT [ms] <a href="#">[L]</a>	Consecutive Drops <a href="#">[L]</a>	ARQ Recovered <a href="#">[L]</a>
Jitter [ms] <a href="#">[L]</a>	Dropped Packets <a href="#">[L]</a>	ARQ Duplicates <a href="#">[L]</a>
Latency [ms] <a href="#">[L]</a>	Recovered Packets <a href="#">[L]</a>	Overflows <a href="#">[L]</a>
	Not Recovered Packets <a href="#">[L]</a>	Consecutive Not Recovered <a href="#">[L]</a>

Any of the columns in the following table can be sorted in ascending/descending order by clicking on its heading. The Search text box in the upper left-hand side of the screen can be used to search for a specific input stream by entering the desired Stream ID.

Field	Description
<b>Status</b>	Displays the status of input stream. The following colored indicators are displayed: <ul style="list-style-type: none"><li>Green = Connected</li></ul>

## Broadcaster Inputs

---

### Viewing Stream Statistics

---

- Yellow = Connecting
  - Red = Disconnected with an error condition
  - Blue = Offline with no connection attempts
  - Grey = Offline and stopped
- 

<b>ID</b>	Displays the unique ID of the stream, as defined in the Zixi Broadcaster.
<b>Type</b>	Displays the type of input stream.
<b>Source</b>	Displays the source host and port number.
<b>Bitrate [kbps]</b>	Displays bitrate of the connected input stream.
<b>Uptime</b>	Displays the amount of time the stream has been active.
<b>TR 101 209</b>	Displays if the Analyzer is active and the streams that are being analyzed according to the TR101 protocol. <b>P1</b> and <b>P2</b> are error priorities.
<b>Error</b>	Displays connection and transmission errors.
<b>Outputs</b>	Displays the amount of outputs currently configured for this input stream.
<b>Actions</b>	Provides access to various actions to be performed on this input stream (see <a href="#">Broadcaster Actions on Input Streams</a> ).

---

## Viewing Stream Statistics

You can view all the accumulative statistics for a connected input stream in real-time. It is possible to reset the counters by clicking **Reset**. The statistics may differ between the various stream types.

### ➔ To View Stream Statistics:

1. On the Inputs page of Zixi Broadcaster, click the desired input stream for which you want to view statistics for. The stream's statistics are displayed on the bottom of the screen.

<b>Field</b>	<b>Description</b>
<b>Bitrate (kbps)</b>	Displays the current bitrate of the input stream.
<b>Uptime</b>	Displays the amount of time the stream has been active.
<b>Reconnections</b>	Displays the number of reconnection attempts.

---



## Broadcaster Inputs

---

### Viewing Stream Statistics

---

---

<b>Total Packets</b>	Displays the total number of packets that have been transmitted.
<b>Packet Rate</b>	Displays the current number of packets per second.
<b>Remuxing Resets</b>	Remuxing resets can occur when remuxing to CBR (i.e value of 'Remux to bitrate' field is set). In this case, the value is not high enough to remux the video+audio+other PIDs data. A possible solution is to increase value of the 'Remux to bitrate' parameter.
<b>Remuxing Overflows</b>	This indicates that the multiplexer internal queues are full.
<b>Smoothing Drops</b>	The Smoothing feature uses a buffer to time packets according to the stream's clock. This is useful when there are bursts in the rate in which the stream arrives. The smoothing drops indicator indicates that the stream is too "bursty".
<b>RTT (ms)</b>	Displays the Round Trip Time (RTT) from the source (in milliseconds).
<b>Jitter (ms)</b>	Displays the current Jitter, measured in milliseconds.
<b>Latency (ms)</b>	Displays the stream latency configured on the source Feeder.
<b>Total Packets</b>	Displays the total number of packets that have been transmitted.
<b>Packet Rate</b>	Displays the current number of packets per second.
<b>Packet Loss %</b>	Displays current percentage of the packets that were dropped on route from the source.
<b>Dropped Packets</b>	Displays the total number of packets dropped between the Feeder and the Broadcaster since the beginning of the stream.
<b>Recovered Packets</b>	Displays the number of dropped packets that have been recovered since the beginning of the stream.
<b>Non Recovered Packets</b>	Displays the number of non-recovered packets.
<b>FEC Packets</b>	Displays the number of FEC packets transmitted.
<b>FEC Recovered</b>	Displays the number of FEC packets recovered.
<b>ARQ Requests</b>	Displays the number of requests for retransmission of dropped packets made with ARQ.
<b>ARQ Recovered</b>	Displays the number of dropped packets recovered via ARQ.
<b>ARQ Duplicates</b>	Displays the number of duplicate recovery packets received via ARQ.
<b>Overflows</b>	The number of packets lost due to buffer overflow.

---

---

## Broadcaster Inputs

---

### Viewing Stream Statistics

---

---


**Note:** Overflow is usually the result of the Max Bitrate being too low.

---

---

## Searching for Input Streams

### ➔ To Search for an Input Stream:

1. In the **Inputs** page of Zixi Broadcaster, type in the specific input stream name in the search text box on the left-hand top of the screen and click .
2. The **Input** screen displays the requested output stream(s).

## Broadcaster Actions on Input Streams


Every stream has a shortcut button that can be used to perform relevant actions on the stream. The following actions can be performed through the Actions button:

- Adding Analysis to the Input Stream
- Recording the Input Stream
- Stopping/Starting the Input Stream
- Deleting Input Streams from Broadcaster
- Editing Input Settings or Duplicating Input Streams
- Transcoding an Input Stream
- Playing Input Streams with Players
- Add Output
- Traceroute

## Stopping/Starting the Input Stream

You can start or stop the input stream at any time.

### ➔ To Stop/Start the Input Stream:

1. On the **Inputs** page of Zixi Broadcaster, click the desired row of the input stream that you want to stop.
2. Click the  button on the right end of the selected row and then **Stop**. The selected row which was a Connected Input Stream is no longer displayed and appears below as **Offline**.  
To continue the Input Stream click **Start** from the same menu.

## Editing Input Settings or Duplicating Input Streams


### ➔ To Edit the Configuration of an Input Stream:

1. In the Zixi Broadcaster **Inputs** page, click the desired input stream that you want to edit.

## Broadcaster Inputs


---

### Broadcaster Actions on Input Streams

2. Click  button on the right end of the selected row and then **Edit**. The Edit input stream name window is displayed. You can modify any of the settings.


Click **OK**. The new settings will take effect.

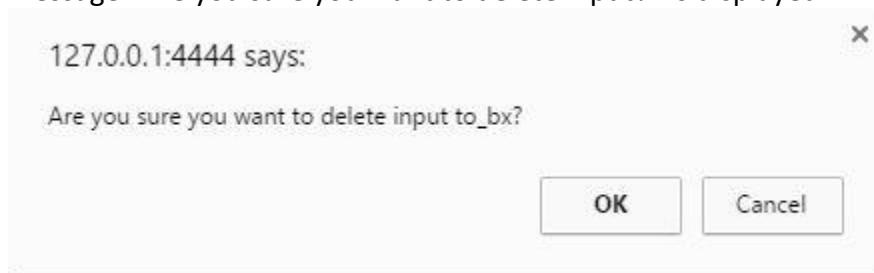
#### ➔ To Duplicate the Configuration of an Input Stream:

1. In the Zixi Broadcaster **Inputs** page, click the desired input stream that you want to duplicate.
2. Click  button on the right end of the selected row and then **Duplicate**. The **Duplicate input stream name** window with the settings of the original input stream is displayed. You can modify any of these settings.
3. Click **OK**. The duplicate input stream with its new settings will be created.

## Deleting Input Streams from Broadcaster

#### ➔ To Delete an Input Stream from Broadcaster:

1. In the Zixi Broadcaster **Inputs** page, click the desired input stream that you want to delete.
2. Click the  button on the right end of the selected row and then **Delete**. The message "Are you sure you want to delete input?" is displayed.



3. Click **OK**. The selected row which was a Connected Input Stream will be deleted.

## Recording the Input Stream

You can record an input stream and store it as .TS file. The duration of the time the recording will be kept on the server is defined in the Input Stream configuration. It is possible to record to an external destination by replacing the default storage location with a symbolic link. To learn more, see [Using an External Storage](#) section.


#### ➔ To Record the Input Stream:

1. In the Zixi Broadcaster **Inputs** page, click the desired input stream that you want to record.

## Broadcaster Inputs

---

### Broadcaster Actions on Input Streams

2. Click the  button on the selected row and then choose **Record**. A small red dot appears in the green circle at the left-hand side of the selected row indicating that the input stream is being recorded.
3. Clicking **Stop Recording** will stop the recording of the Input Stream and the newly recorded file will appear in the designated file directory in the **Files** page.



## Broadcaster Inputs

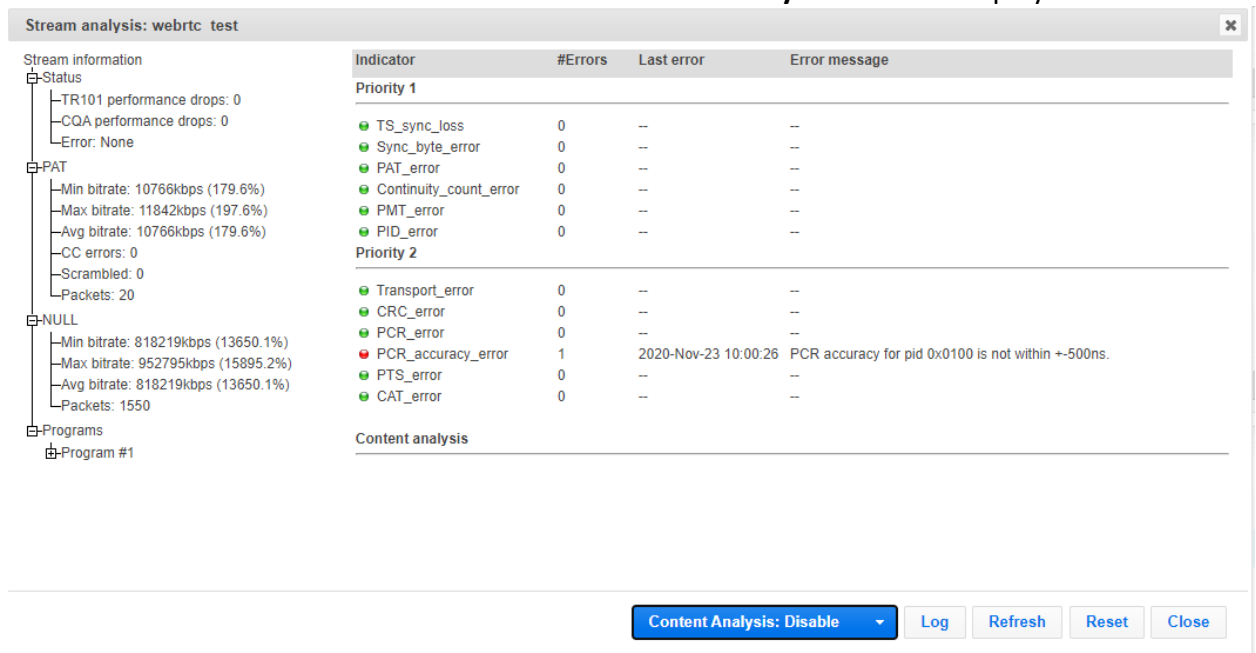
### Broadcaster Actions on Input Streams

## Adding TR101 Analysis to the Input Stream

A TR101 analysis can be added to the input stream for the monitoring of the TS while in operation. The analysis can be used to check the integrity of the TS and the most important elements of the stream. The analysis includes priority 1 and 2. To learn more about TR101 analysis see [TR 101 290 Technical Report](#). For more information on Zixi's TR101 analysis, see [Viewing TR101 Analysis](#).

### ➔ To Add Analysis to the Input Stream:

1. In the Zixi Broadcaster **Inputs** page, click the stream for which you want to analyze from the input streams UI.
2. Click the  button on the right end of the selected row and then select **Analyze**. The statuses of a **P1** (high priority error) and **P2** (low priority error) are indicated by an adjacent green or red button displayed in the *TR 101 290* column. You can view the detailed analysis of the P1 and P1 errors by clicking on the stethoscope icon  shown in the TR 101 290 column. The **Stream Analysis** screen is displayed.



The screenshot shows the 'Stream analysis: webrtc test' window. On the left is a tree view of stream information, and on the right is a table of error indicators.

Indicator	#Errors	Last error	Error message
<b>Priority 1</b>			
TS_sync_loss	0	--	--
Sync_byte_error	0	--	--
PAT_error	0	--	--
Continuity_count_error	0	--	--
PMT_error	0	--	--
PID_error	0	--	--
<b>Priority 2</b>			
Transport_error	0	--	--
CRC_error	0	--	--
PCR_error	0	--	--
PCR_accuracy_error	1	2020-Nov-23 10:00:26	PCR accuracy for pid 0x0100 is not within +-500ns.
PTS_error	0	--	--
CAT_error	0	--	--
<b>Content analysis</b>			

At the bottom of the window, there are buttons: 'Content Analysis: Disable' (with a dropdown arrow), 'Log', 'Refresh', 'Reset', and 'Close'.

3. The following can be done from the **Stream Analysis** screen:
  - **Content Analysis** – shows errors in the audio and video content (i.e. Frozen video, Blank picture, Silent audio, Low video quality and Audio clipping).
  - **Log**- start sending the events of this stream to the event log ('stop logging' will stop the reporting of events).
  - **Refresh** – refreshes the data that is displayed.
  - **Reset** – resets all the field's values to zero.

On the left-hand side of the screen is a stream information tree that displays accumulative statistics regarding the stream. Click the + icons to expand the tree.

## Broadcaster Inputs

---

### Broadcaster Actions on Input Streams

4. Clicking **Stop Analysis** will discontinue the action.

## Transcoding an Input Stream

Zixi Broadcaster offers live transcoding, enabling users to create multiple bitrate streams from a single input stream.

Supported Video and Audio formats include:

- Video Decoding – MPEG2, H.264/AVC, H.265/HEVC
- Video Encoding – MPEG2, H.264/AVC, H.265/HEVC
- Audio Decoding – AAC, MPEG-1 Audio Layer I / II / III, MPEG-2 Audio Layer I / II / III, AC-3 (Dolby Digital)



For AC-3, which can contain up to six discrete channels of sound, the transcoder only supports the left and right channels and will ignore the other channels.

- Audio Encoding – AAC, AAC-HE (High Efficiency) and AAC-HEv2; Pass through (no encoding) – any audio codec (MPEG1/2 / AAC / AC-3, E-AC-3 (Dolby Digital Plus)).

The transcoder comes with pre-set common video profiles:


1080i60 / 1080i59.94, 1080p30 / 1080p25, 720p60, 720p50 / 720p30, 576i60 / 576p25, 480i60 / 480p29.97

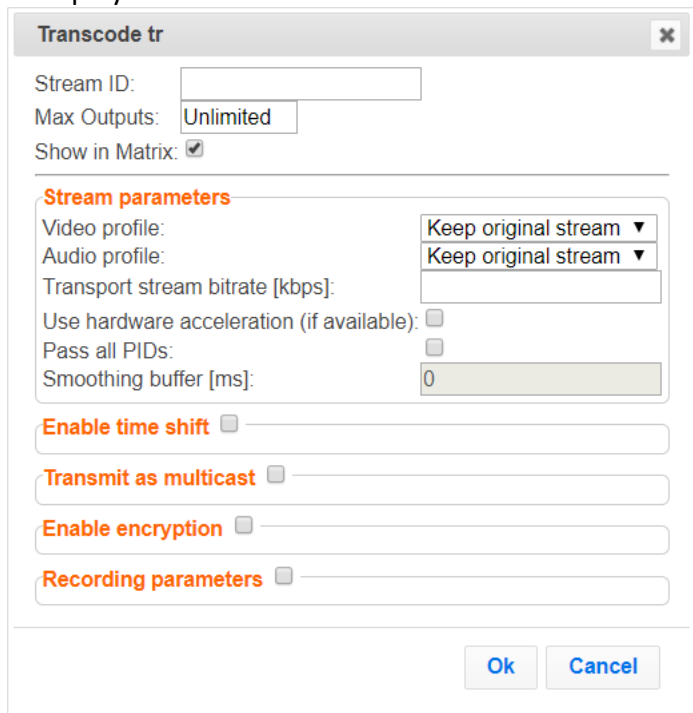
In order to transcode an input stream from the Inputs screen you must first configure the transcoding profiles in the Transcoder screen. For more information, see [Transcoder](#) section

## Broadcaster Inputs

### Broadcaster Actions on Input Streams

#### ➔ To Transcode an Input Stream:

1. In the Zixi Broadcaster **Inputs** page, click the desired input stream that you want to transcode.
2. Click  button on the right end of the selected row and then **Transcode**. The **Transcode <input stream name>** window with the settings of the selected input stream is displayed.



3. Enter the unique **Stream ID.**, which will correspond to the new transcoded input that will be added.
4. Under Stream Parameters, in the **Video profile** and **Audio profile** fields, do one of the following:
  - Select the desired Video Profile and/or Audio Profile that you have configured.
  - Select **Keep original stream** - if you have not configured a video or audio profile or do not want it transcoded or you want to set specific transcoding parameters for the specific stream. In this case the video or audio input will not be transcoded.
  - Select **Remove stream** – you can use the transcoder to remove the video or audio track from the stream.
5. Fill in additional fields (see table below for additional parameters).
6. Click **OK**. The transcoded input stream is created and appears in an embedded window labeled Transcoded sources under the original input stream from which it was created.

---

Parameter	Description
-----------	-------------

---

---



## Broadcaster Inputs

---

### Broadcaster Actions on Input Streams

---

<b>Video Profile</b>	Select an existing profile, select <b>Keep original stream</b> if you do not wish to transcode the video stream, or select <b>Remove stream</b> if you want to remove the video track.
<b>Audio Profile</b>	Select an existing profile, select <b>Keep original stream</b> if you do not wish to transcode the audio stream, or select <b>Remove stream</b> if you want to remove the audio track.
<b>Transport Stream Bitrate (Kbps)</b>	Specify the bitrate of the transport stream. This parameter is optional. If you do not specify a TS Bitrate, the stream will be VBR. If you specify a bitrate, the stream will be CBR.
<b>Use Hardware Acceleration (if available)</b>	Select this checkbox to use hardware acceleration component (e.g. NVIDIA or Intel) if it has been installed.
<b>Pass all PIDs</b>	Select this checkbox to pass-through all non-transcoded PIDs.
<b>Smoothing Buffer (ms)</b>	<p>Specify how much of the transcoded stream (in milliseconds) to store before transmitting it back to the server in order to prevent bursts.</p> <p>If the <b>Transport stream bitrate [kbps]</b> field is empty, the smoothing buffer is disabled.</p> <p>If the <b>Transport stream bitrate [kbps]</b> field is NOT empty, it is recommended to enter around 200ms, which should be enough to prevent bursts.</p>
<b>Enable time shift</b>	
<b>Maximum delay [hh:mm:ss]</b>	Specify the time of the delayed transmission of this stream. Maximum: 24 hours.
<b>Transmit as multicast</b>	Selecting this checkbox enables the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default, Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling Multicast Pull in the Settings. This setting requires enabling Multicast Pool (Settings > Multicast Pool).
<b>Multicast Only</b>	Selecting this checkbox will force the transmission of this stream only in multicast.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a> .
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).
<b>Encryption key (for encryption enabled)</b>	Click <b>Generate</b> to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.

---

## Broadcaster Inputs

### Broadcaster Actions on Input Streams

<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.
<b>Destination Type</b>	Select the radio button for the desired storage type and then fill in the relevant parameters. <ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk.</li><li>• <b>Record to S3</b> – the file is saved to your AWS S3 Bucket.</li></ul>
<b>Record to Disk Parameters</b>	
<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours
<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.
<b>Record to S3 Parameters</b>	
<b>File Name Template</b>	Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.

## Broadcaster Inputs

---

### Broadcaster Actions on Input Streams

---

The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:

- %S=stream id - optional.
- %Y=year - optional
- %M=month - optional
- %D=day - optional
- %T=HH.MM.SS creation time - mandatory

For example, the following pattern %Y\_recording%T.ts will generate the following file name 2017\_recording15:32:35.ts.

---

<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
---	--

---

<b>URL</b>	Specify the URL of the S3 bucket.
------------	-----------------------------------

---

<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
--------------------------------------	---

---

<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
-------------------	--

---

<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.
-------------------	--

---

## Add Output


This is an alternative method (shortcut) to adding an output to a specific input stream instead of configuring it through the **Outputs** screen. The available types of outputs are:

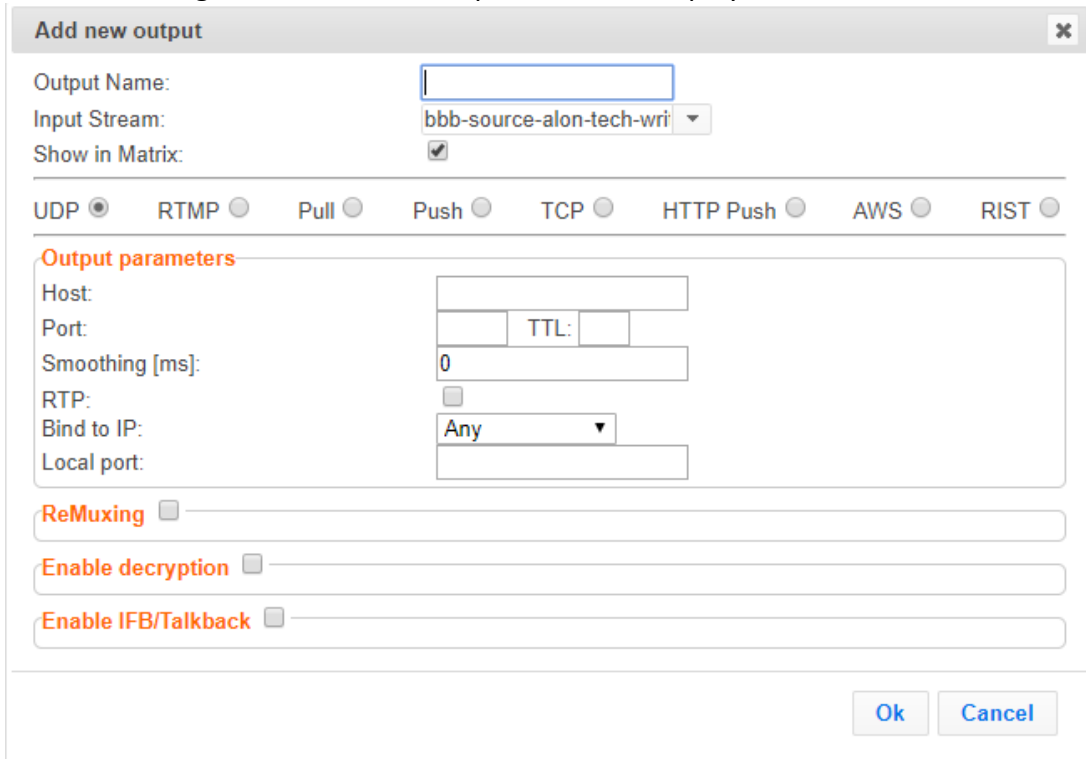
- **UDP** – Target is a UDP or RTP stream, over unicast or multicast.
- **RTMP** – Real Time Messaging Protocol streams can be played on systems that accept this protocol as input. Originally designed to enable Adobe® Flash® streams to be sent between a server and a client, RTMP has become a popular protocol for streaming into content-delivery networks (CDNs).
- **Pull** – Pull streams are Zixi protocol streams destined for Zixi Receivers. In this case, the Zixi Broadcaster can offer status and statistics for a stream that is being pulled from it by another device.
- **Push** – Push streams are Zixi protocol streams destined for other Zixi Broadcasters.
- **TCP** – The TCP output can push the stream directly to a TCP socket or to an HTTP server.
- **HTTP Push** – The HTTP Push output stream can be encapsulated in HLS or DASH. The source of the output stream is a predefined adaptive group that has been configured in the [Adaptive Groups](#) screen. The configuration of the stream includes the option to delete outdated stream fragments on the server. Output streams can be pushed to one Zixi Broadcasters or Zixi Receiver.
- **AWS** – The HTTP output stream (HLS or DASH) can be pushed to an AWS S3 Bucket or too an Elemental MediStore. The native S3 support includes the ability to configure the S3 authentication credential, including the Region, Access Key, and Secret Key.
- **RIST** – The RIST output stream can be pushed to an external target using Simple or Main Profile over unicast or multicast. The output stream can include Forward Error Correction (SMPTE-2022 FEC).

## Broadcaster Inputs

### Broadcaster Actions on Input Streams

#### ➔ To Add Output:

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
2. Click the Settings button , and then click **New Output**. The **New Output** window with the settings of the selected output stream is displayed.



**Add new output**

Output Name:

Input Stream: bbb-source-alon-tech-wri

Show in Matrix:

UDP  RTMP  Pull  Push  TCP  HTTP Push  AWS  RIST

**Output parameters**

Host:

Port:  TTL:

Smoothing [ms]:

RTP:

Bind to IP: Any

Local port:

ReMixing

Enable decryption

Enable IFB/Talkback

Ok Cancel

3. Follow the specific instructions of each output type from one of the following sections:
  - **Adding UDP Outputs**
  - **Adding RTMP and RTMPS Outputs**
  - **Adding Pull Outputs**
  - **Adding Push Outputs**
  - **Adding TCP Outputs**
  - **Adding HTTP Push Outputs**
  - **Adding AWS Outputs**
  - **Adding RIST Outputs**


## Playing Input Streams with Players

You can preview the input streams through the following video players:

- VLC Player
- Flash Player
- Shoutcast
- WebRTC


In order to preview an input stream you must first configure its output. See Broadcaster Outputs section for more details.

### ➔ To Play Input Streams with a VLC Player:

1. In the Zixi Broadcaster **Inputs** page, click the desired input stream that you want to play with VLC.
2. Click the  button on the right end of the selected row and then **Play with VLC**. The **Link Description** window with the directory path for the desired file is displayed.
3. Click **Run VLC**. The selected file is downloaded to the computer.
4. Double-click on the newly downloaded file. This configured file runs using the VLC player and appears on a new output line on the **Output** screen wherein it can be monitored. Alternatively, copy the path and send it to the user so it can be entered in the Network URL of the VLC player.


### ➔ To Play Input Streams with a Flash Player:

Flash Player enables playing the stream on a native browser with the Flash player.

1. In the Zixi Broadcaster **Inputs** page, click the desired input stream that you want to play with Flash Player.
2. Click the  button on the right end of the selected row and then **Play with Flash**. The **Link Description** window with the directory path and proxy path for the desired file is displayed.
3. Click **Run** to view the .flv file that appears on the first line of the window or **Run Accelerated** to run the file from the proxy server with better results.

### ➔ To Play Input Streams with Shoutcast:

Shoutcast enables playing an audio file on a native browser.

1. In the Zixi Broadcaster **Inputs** page, click the desired input stream that you want to play with Shoutcast.
2. Click the  button on the right end of the selected row and then **Play with Shoutcast**. The **Link Description** window will open with the directory path for the desired file.


## Broadcaster Inputs

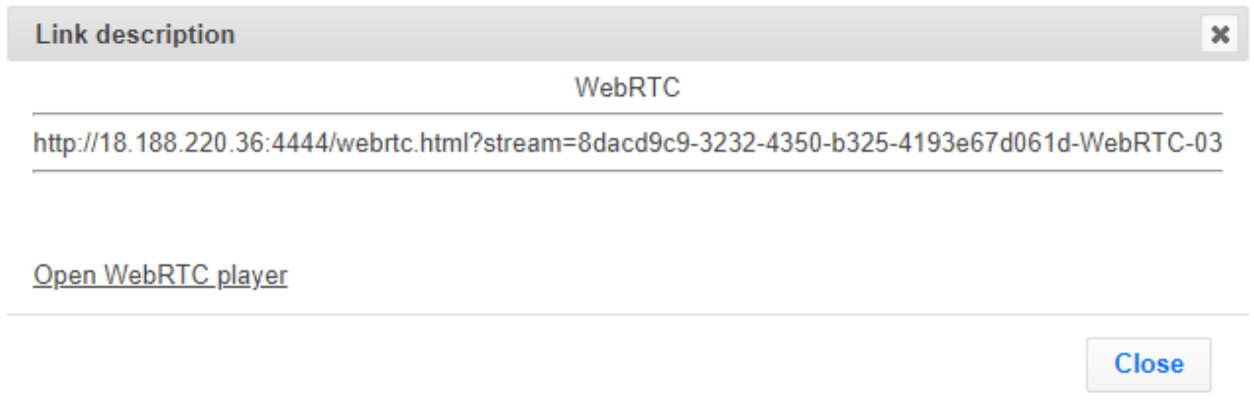
---

### Broadcaster Actions on Input Streams

#### ➔ To Play Input Streams with WebRTC:

WebRTC enables playing a video file on a native browser without the need for plug-ins or extensions.

1. In the Zixi Broadcaster **Inputs** page, click the desired WebRTC input stream that you want to play.
2. Click the  button on the right end of the selected row and then **Play WebRTC**. The **Link Description** window will open with the directory path for the desired file and a link to open a browser window to play the stream.



Copy the link or click the **Open WebRTC player** link to view the stream.


## Broadcaster Inputs

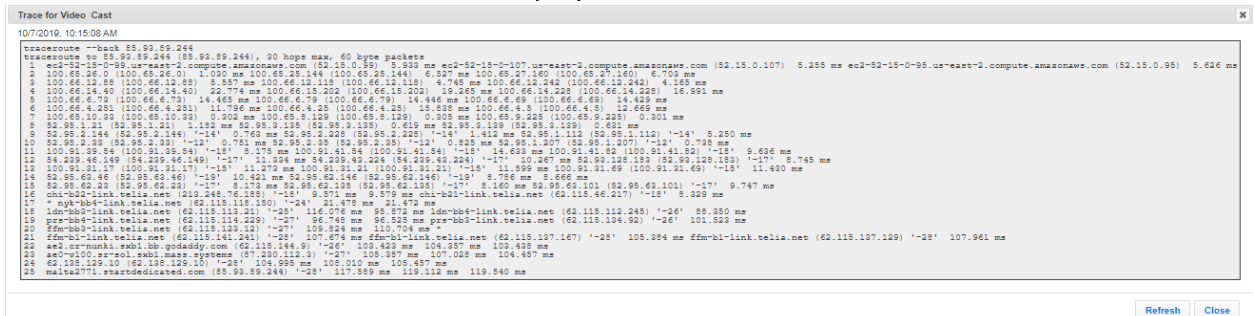
### Broadcaster Actions on Multiple Inputs

## Viewing Traceroutes

For streams sent from a Zixi Feeder (v13 and above) to a Zixi Broadcaster (v13 and above), you can view traceroute data for the stream.

### ➔ To view traceroute data:

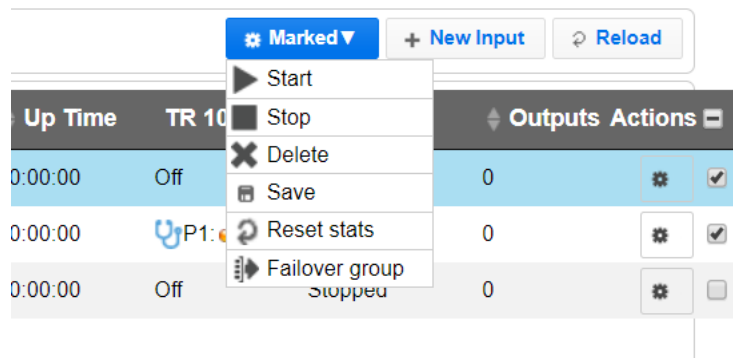
1. On the **Inputs** page of Zixi Broadcaster, click on the row of the input stream for which you want to view traceroutes.
2. Click the  button on the right end of the selected row and then select **Trace**.
3. The traceroute data for this stream is displayed.



4. You can refresh the Traceroute data at any time by clicking the **Refresh** button.

## Broadcaster Actions on Multiple Inputs

In the Inputs screen, you can perform actions on multiple or all input streams at once by selecting them using the check boxes and then clicking the Marked drop-down menu.



The actions include the following:

## Starting and Stopping Multiple Input Streams

You can Start/Stop multiple streams.

### ➔ To Start/Stop multiple Input Streams:

1. In the Zixi Broadcaster Inputs screen, select the relevant Inputs by selecting the checkboxes



## Broadcaster Inputs

---

### Broadcaster Actions on Multiple Inputs

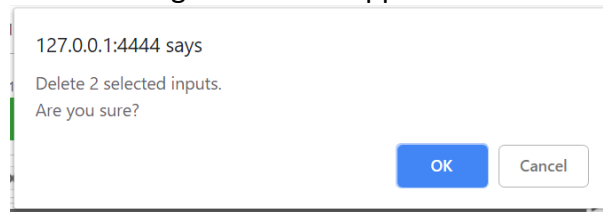
2. Click the **Marked** drop-down menu and then click **Stop** or **Start**.  
The selected input streams will be started/stopped.

## Deleting Multiple Input Streams

### ➔ To Delete Multiple Input Streams:

1. In the Zixi Broadcaster Inputs screen, select the relevant Inputs by selecting the checkboxes.
2. Click the **Marked** drop-down menu and then click **Delete**.

The following notification appears:



3. Click **OK**.

## Saving Multiple Input Streams

### ➔ To Save Multiple Input Streams:

1. In the Zixi Broadcaster Inputs screen, select the relevant Inputs by selecting the checkboxes.
2. Click the **Marked** drop-down menu and then click **Save**.
3. All changes to the input streams are saved.

## Resetting Stats for Multiple Input Streams

### ➔ To Reset Stats for Multiple Input Streams:

1. In the Zixi Broadcaster Inputs screen, select the relevant Inputs by selecting the checkboxes.
2. Click the **Marked** drop-down menu and then click **Reset Stats**.  
All accumulated stream statistics are reset.

## Creating a Failover Group

Zixi features hitless failover for inputs, enabling uninterrupted streaming when switching from one source to another. The hitless failover configuration involves grouping two or more inputs in a "Failover Group". If the inputs are identical (i.e. two or more binary-identical streams with synchronized RTP headers), whenever one of them is disrupted or interrupted completely, Zixi

## Broadcaster Inputs

### Broadcaster Actions on Multiple Inputs

Broadcaster will seamlessly use the packets of the other input without any disruptions by "merging" the streams. The merging of the streams can be achieved using one of the following methods:

- **RTP headers (SMPTE 2022-7)** – if the sources are RTP and the RTP headers were maintained, Zixi will use the SMPTE 2022-7 to compare and merge the two streams in order to allow hitless failover. This option is the most recommended as it will use the sequence and the timestamp in the packets to compare which packets are missing. This method is more accurate than methods that compare the entire payload or methods that use sophisticated DNA sequencing algorithm, which are more error prone, as the payload may have multiple matches.
- **RTP headers or TS content** – if the sources do not have RTP headers, Zixi will try to merge using the MPEG-TS content itself.



If the streams are not identical, the failover process will not be seamless and may take up to a second.

The following procedure describes how to configure inputs as a Failover Group in Zixi Broadcaster.

#### ➔ To create a Failover Group:

1. On the Input screen of Zixi Broadcaster, select the checkboxes of the relevant identical input streams.
2. At the top of the inputs list, click the **Marked** drop-down menu and then click **Failover Group**.

The screenshot shows the Zixi Broadcaster interface with the 'Inputs' tab selected. The top navigation bar includes STATUS, INPUTS, OUTPUTS, ADAPTIVE, VOD, FILES, TRANSCODER, SETTINGS, EVENT LOG, MATRIX, and LOGOUT. The main content area displays a table of inputs with columns for Status, ID, Type, Source, Bitrate[kbps], Up Time, TR 101 290, and Error. A dropdown menu is open over the table, showing options: Start, Stop, Delete, Save, Reset stats, and Failover group. The 'Failover group' option is highlighted. The table contains four rows of input data.

Status	ID	Type	Source	Bitrate[kbps]	Up Time	TR 101 290	Error
Connected	test0	Push	10.9.33.55	26112	00:52:46	P1: P2:	None
Connected	test1	Push	10.9.33.55	25609	00:52:46	P1: P2:	None
Connected	test10	Push	10.9.33.55	23065	00:52:46	P1: P2:	None
Connected	test100	Push	10.9.33.55	24047	00:52:45	Off	None

## Broadcaster Inputs

### Broadcaster Actions on Multiple Inputs

3. The **Create a Failover Group** window opens:

**Create failover group**

Stream ID:

Max Outputs:

Show in Matrix:

**Stream parameters**

Search window[ms]:

Max bitrate [kbps]:

Merge mode:

**Candidates**

Stream	Backup	
<input type="text" value="bloom-shift"/>	<input type="checkbox"/>	<input type="button" value="+"/> <input type="button" value="x"/>
<input type="text" value="jelly"/>	<input type="checkbox"/>	<input type="button" value="+"/> <input type="button" value="x"/>

Enable time shift

Transmit as multicast

Recording parameters

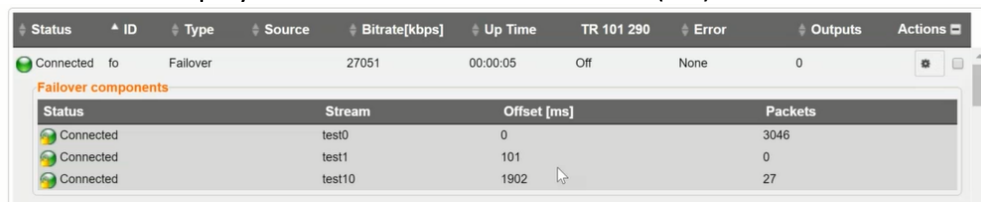
4. In the **Stream ID** field, type a new name for Failover Group. This name will appear in the list of Inputs.
5. In the **Max Outputs** field, type the number of desired concurrent outputs that can use the **Failover Group**. Default: Unlimited.
6. In the **Search Window** field, type the timeframe in which the system can search for the relevant packets in a matching stream. If there is a latency difference between the streams in the group, the **Search Window** value should be higher than the latency difference between the streams. In case the **Merge Mode** is set to **No Merge**, the **Search Window** time specifies how long the stream will wait for packets before attempting to switch between streams.
7. In the **Max Bitrate** field, specify the maximum bitrate of the stream. Default: 8000. This is used for internal memory allocation and will not affect the transmitted bitrate. Recommended: double the bitrate of the highest bitrate stream.
8. In the **Merge mode** field, select one of the following:
  - **No merge** – select this option if you want Zixi to treat the streams as different and not try to merge the streams. In this case, the time period specified in the **Search Window** field defines how long the stream will wait for packets before attempting to switch between streams.

## Broadcaster Inputs

### Encrypting an Input Stream in Zixi Broadcaster

- **RTP headers (SMPTE 2022-7)** – select this option if the sources are RTP and the RTP headers were maintained. In this case, Zixi will use the SMPTE 2022-7 to compare and merge the two streams in order to allow hitless failover. This option is the most recommended as it will use the sequence and the timestamp in the packets to compare which packets are missing. This method is more accurate than methods that compare the entire payload or methods that use sophisticated DNA sequencing algorithm, which are more error prone, as the payload may have multiple matches.
  - **RTP headers or TS content** – select this option if you are not sure if the sources do have RTP headers. In this case, Zixi will try to merge using RTP and if not, using the MPEG-TS content itself.
9. Under **Candidates**, if you want to configure one or more of the streams as back up, select the checkbox next to the relevant stream. Backup streams will be used only when all the "Primary" streams are down.
  10. Configure additional parameters as required.
  11. Click **OK**.

The Hitless Failover Group is added to the list of inputs, listing the sub-streams that are part of the group. The system will automatically detect the offset between the sub-streams and display it in milliseconds under Offset (ms).



The screenshot shows a table titled "Failover components" with the following data:

Status	Stream	Offset [ms]	Packets
Connected	test0	0	3046
Connected	test1	101	0
Connected	test10	1902	27

## Encrypting an Input Stream in Zixi Broadcaster

You can encrypt a specific input stream using static key AES scrambling encryption (AES 128, AES 192, AES 256). If the stream has been encrypted at the source, it can be encrypted by Zixi Broadcaster only after its decryption (See: [Decrypting an Input Stream in Zixi Broadcaster](#)). The stream encryption prevents any further processing, such as analysis, transcoding, etc. For more information, see [TRANSCODER](#) section.

### ➔ To encrypt an input stream in Zixi Broadcaster:

1. In **Inputs** screen, click **+New Input**.
2. Select the relevant input type.
3. Fill in the stream parameters.
4. At the bottom, select the **Enable Encryption** checkbox.
5. In the **Encryption Type** field, select **AES 128**, **AES 192**, **AES 256** for fixed encryption.
6. In the **Encryption Key** field, enter your own key or click **Generate** to generate a key by the system. The exact key should be used to decrypt the stream on the other end.

## Broadcaster Inputs

### Decrypting an Input Stream in Zixi Broadcaster

**Add new input stream** [X]

Stream ID:

Max Outputs:

Show in Matrix:

Push  Pull  UDP  Delayed  File  RTMP  RTSP  HLS  TS over HTTP

**Stream parameters**

Password:

Latency [ms]:

Point to point:

ID of high priority source:

Disable P2P fallback:

**Enable time shift**

**Transmit as multicast**

**Decrypt stream**

**Enable encryption**

Stream will be encrypted after decryption (where applicable).  
Encryption will prevent stream processing (e.g. transcoding).

Encryption type:

Encryption key:

**Recording parameters**

7. Click **OK**.

## Decrypting an Input Stream in Zixi Broadcaster

Any stream that has been encrypted using standard AES static key can be decrypted in Zixi Broadcaster. You can configure the decryption of a specific stream by entering the decryption key.

### ➔ To decrypt an input stream in Zixi Broadcaster:

1. In **Inputs** screen, click **+New Input**.
2. Select **Push** or **Pull**.
3. Fill in the input details.
4. At the bottom, select the **Decrypt Stream** checkbox.
5. In the **Decryption Type** field, select **AES 128**, **AES 192**, **AES 256** for fixed key.
6. In the **Decryption Key** field, enter the exact same key that was used to encrypt the stream.

## Broadcaster Inputs

### Demuxing an MPTS Stream

**Add new input stream** [X]

Stream ID:

Max Outputs:

Show in Matrix:

Push  Pull  UDP  Delayed  File  RTMP  RTSP  HLS  TS over HTTP

**Stream parameters**

Password:

Latency [ms]:

Point to point:

ID of high priority source:

Disable P2P fallback:

**Enable time shift**

**Transmit as multicast**

**Decrypt stream**

\*Automatic encryption does not require manual settings.

Decryption type:

Decryption key:

**Enable encryption**

**Recording parameters**


Ok Cancel

7. Click **OK**.

## Demuxing an MPTS Stream

MPTS (Multi Program Transport Streams) streams can be demuxed into single SPTS (Single Program Transport Streams) streams. MPTS demuxing is available for Pull, Push, UDP, and File input streams. From the various programs in the MPTS stream, you can select specific programs, which will be turned into individual SPTS input streams.

### ➔ To demux an MPTS stream:

1. Create a Pull/Push/UDP/File input stream using an MPTS stream.
2. In the **Inputs** page, click the newly created input stream that you want to demux.
3. Click the action  button on the right end of the selected row and then **Edit**. The Edit input stream name window is displayed.
4. Select the **Demux MPTS** checkbox.  
A list of included programs is displayed.

## Broadcaster Inputs

### Demuxing an MPTS Stream

Program#	Name	Provider	Assigned input
1			
4			
5			
6			
7			
8			
9			
10			
11			
12			
..			

5. Click **Add Program** on the program that you want to create a separate SPTS input stream.

An **Add Input** screen opens with the new SPTS input details, including a suggested **Stream ID**, the original MPTS as a Source Stream and the relevant **Program** number.

Program #4 of stream MPTS\_example

Stream ID: MPTS\_example\_program

Max Outputs: Unlimited

Show in Matrix:

**Stream parameters**

Source stream: MPTS\_example

Program: 4

Max bitrate [kbps]:

Enable time shift

Transmit as multicast

Enable encryption

Recording parameters

Ok Cancel

6. In the **Max Bitrate** field, specify the maximum bitrate of the stream. This is used for internal memory allocation and will not affect the transmitted bitrate. Recommended: double the bitrate of the highest bitrate stream.
7. Configure additional parameters as necessary.
8. Click **OK**.

The channel is assigned an input:

## Broadcaster Inputs

### Viewing TR101 Analysis

Push  Pull  UDP  Delayed  File  RTMP  RTSP  HLS  TS over HTTP

**Stream parameters**

Path:  ...

**Enable time shift**

**Transmit as multicast**

**Enable encryption**

**Demux MPTS**

Program#	Name	Provider	Assigned input	
1				+ Add program
4			MPTS_example_program_4	<input type="text"/> <input type="text"/>
5				+ Add program
6				+ Add program
7				+ Add program

- Repeat steps 5-8 to add more programs.
- Click **OK**.

The newly created MPTS input stream and the related SPTS input streams will appear in the Inputs list.

STATUS INPUTS OUTPUTS ADAPTIVE VOD FILES TRANSCODER SETTINGS EVENT LOG MATRIX LOGOUT


Inputs

Status	ID	Type	Source	Bitrate[kbps]	Up Time	TR 101 290	Error	Outputs	Actions
Connecting	mpts2	Pull	demo.zixi.com:2088	0	00:00:00	Off	No Source	0	<input type="button" value="x"/> <input type="button" value="p"/>
Connected	MPTS_example	File	Curtis/MC_HD_Audio_Channels.ts	19025	04:01:31	Off	None	0	<input type="button" value="x"/> <input type="button" value="p"/>
Connected	MPTS_example_program...	SPTS	MPTS_example.Program#4	497	00:07:15	Off	None	0	<input type="button" value="x"/> <input type="button" value="p"/>
Connected	Music_Choice_MPTS	File	Curtis/MC_HD_Audio_Channels.ts	19010	136:38:52	<input checked="" type="checkbox"/> P1: <input checked="" type="checkbox"/> P2: <input checked="" type="checkbox"/>	None	1	<input type="button" value="x"/> <input type="button" value="p"/>

## Viewing TR101 Analysis

TR101 analysis can be added to an input stream for the monitoring of the TS while in operation. The analysis can be used to check the integrity of the TS, as well as perform content analysis to identify frozen video, blank picture, silent audio, low video quality and audio clipping. The TR101 analysis covers priority 1 and 2 messages.

To enable the TR101 analysis follow the instructions in the [Adding TR101 Analysis to the Input Stream](#) section. After enabling the analysis, the stream will include the P1 (high priority error) and P2 (low priority error) status, indicating is there are P1 or P2 messages.

To open the analysis window, click the stethoscope () icon.

The TR101 Analysis window includes the following elements:



## Broadcaster Inputs

### Viewing TR101 Analysis

The screenshot shows a software interface for stream analysis. On the left, a tree view displays 'Stream information statistics' with categories: Status (TR101 performance drops: 0, CQA performance drops: 0, Error: None), PAT (Min/Max/Avg bitrate: 15kbps (0.3%), CC errors: 0, Scrambled: 0, Packets: 142286), NULL (Min/Max/Avg bitrate: 515kbps (8.6%), Packets: 7662432), and Programs (PMT PID: 4096 (0x1000), Min/Max/Avg bitrate: 15kbps (0.3%), CC errors: 0). The main panel shows a table of indicators with columns for Indicator, #Errors, Last error, and Error message. Indicators are grouped into Priority 1, Priority 2, and Content analysis. At the bottom, there are buttons for 'Content Analysis: All Frames', 'Log', 'Refresh', 'Reset', and 'Close'. Labels on the right side of the image point to 'P1/P2 Errors' (Priority 2 section), 'Content analysis errors' (Content analysis section), and 'Analysis Controls' (bottom buttons).

## Stream Information Statistics Panel

The stream information statistics panel displays the following accumulative statistics about the stream:

Parameter	Description
<b>Status</b>	
<b>TR101 performance drops</b>	
<b>CQA performance drops</b>	Specify the UDP port to be used for streaming the video.
<b>Error</b>	Specify the UDP port to be used for streaming the audio.
<b>PAT</b>	The Program Association Table (PAT), which only appears in PID 0x0000 packets, tells the decoder what programs are in the TS and points to the Program Map Tables (PMT) which in turn point to the component video, audio and data streams that make up the program. If the PAT is missing then the decoder can do nothing, no program is decodable.
<b>Min bitrate</b>	The minimum bitrate of the PAT packets
<b>Max bitrate</b>	The maximum bitrate of the PAT packets

## Broadcaster Inputs

### Viewing TR101 Analysis

<b>Avg bitrate</b>	The average bitrate of the PAT packets
<b>CC errors</b>	Indicates if continuity count errors were found. For this indicator three checks are combined. The preconditions "Incorrect packet order" and "Lost packet" could cause problems for IRD which are not equipped with additional buffer storage and intelligence. It is not necessary for the test equipment to distinguish between these two preconditions as they are logically OR-ed, together with the third precondition, into one indicator.
<b>Scrambled</b>	Indicates if the packet is encrypted.
<b>Packets</b>	The number of packets which are related to the PAT (with the same PID – 0)
<b>NULL</b>	Null (stuffing) packets. In order to ensure that the stream maintains a constant bitrate, a multiplexer may need to insert some additional packets. The PID 0x1FFF is reserved for this purpose. The payload of null packets is all zeroes, and the receiver is expected to ignore its contents. Null packets are indicated by PID 8191 (all 1s)
<b>Min bitrate</b>	The minimum bitrate of the NULL packets
<b>Max bitrate</b>	The maximum bitrate of the NULL packets
<b>Avg bitrate</b>	The average bitrate of the NULL packets
<b>Packets</b>	The number of packets which are related to NULL (with the same PID 8191)
<b>Programs</b>	The Transport Stream can include one program (SPTS) or multiple programs (MPTS). Every program is described by a program map table (PMT). The elementary streams associated with that program have PIDs listed in the PMT. Another PID is associated with the PMT itself.
<b>PMT PID</b>	Indicates the PID number of the program map table PMT
<b>Min bitrate</b>	The minimum bitrate of the packet which is related to the PMT PID
<b>Max bitrate</b>	The maximum bitrate of the packet which is related to the PMT PID
<b>Avg bitrate</b>	The average bitrate of the packet which is related to the PMT PID
<b>CC errors</b>	Indicates if continuity count errors were found in the packets that are related to the PMT PID. For this indicator three checks are combined. The preconditions "Incorrect packet order" and "Lost packet" could cause problems for IRD which are not equipped with additional buffer storage and intelligence. It is not necessary for the test equipment to distinguish between these two preconditions as they are logically OR-ed, together with the third precondition, into one indicator.
<b>PCR PID</b>	Indicated the PID value that includes the program clock reference (PCR). The PCR is used to enable a decoder to present synchronized content, such as audio tracks matching the associated video.

## Broadcaster Inputs

### Viewing TR101 Analysis

<b>PCR</b>	Indicates the program clock reference (PCR) value.
<b>Accuracy</b>	Indicates the difference between the actual PCR value and the value it should have in the TS represented by the byte index for its actual position. For example, if the calculated PCR (derived from adding the value to the previous packet) is the same as the actual PCR value, the Accuracy value will be '0'.
<b>Deviation</b>	Indicates the frequency deviation.
<b>Delay</b>	Indicates the delay in delivery.
<b>OJ max</b>	Indicates the maximal Overall Jitter, which are variation in delays.
<b>OJ Dev</b>	Indicates the deviation in the overall jitter.
<b>Bitrate</b>	The bitrate of the specific PCR PID packets.
<b>Elementary PIDs</b>	<i>Below the Elementary PIDs level are all the different programs that are included in the stream.</i>
<b>Type</b>	The type of program.
<b>Min Bitrate</b>	The minimum bitrate of the packet which is related to the Elementary PID
<b>Max Bitrate</b>	The maximum bitrate of the packet which is related to the Elementary PID
<b>Avg Bitrate</b>	The average bitrate of the packet which is related to the Elementary PID
<b>CC errors</b>	Indicates if continuity count errors were found in the packets that are related to the Elementary PID.
<b>Scrambled</b>	Indicates if the packet is encrypted. Possible values include: Not scrambled, scrambled with even key, scrambled with odd key.
<b>PTS</b>	The Presentation Time Stamp (PTS) that the analyzer has captured on the Element stream. The PTS is only accessible if the TS is not scrambled
<b>DTS</b>	The Decode Time Stamp (DTS) that the analyzer has captures on the Element stream. The Decode Time Stamp (DTS) indicates the time at which an access unit should be instantaneously removed from the receiver buffer and decoded. It differs from the Presentation Time Stamp (PTS) only when picture reordering is used for B pictures. If DTS is used, PTS must also be provided in the bit stream. PTS (or DTS) is entered in the bitstream at intervals not exceeding 700 mS. ATSC further constrains PTS (or DTS) to be inserted at the beginning of each access unit.
<b>Total Bytes</b>	The total amount of bytes that has passed through the current elementary stream. This includes the Transport Stream Headers.
<b>Packets</b>	The total amount of packets that have passed through the current elementary stream.

## Broadcaster Inputs

### Viewing TR101 Analysis

<b>Width</b>	The frame resolution width setting.
<b>Height</b>	The frame resolution height setting.
<b>FPS</b>	The Frames Per Second setting.
<b>Interlaced</b>	Indicates whether the video is interlaced or not.
<b>Chroma</b>	Indicates the Chroma Subsampling.
<b>Frames</b>	Indicates the number of frames. This indicator is enabled when content analysis is turned on.
<b>Content</b>	Indicates the status of the content. This indicator is enabled when content analysis is turned on. Possible values: <ul style="list-style-type: none"><li>• Live – indicates that there is a difference between two consecutive frames within a period of 30 seconds.</li><li>• Frozen video – indicates that there is a single frame that is displayed (frozen).</li><li>• Blank picture – the entire frame is in one color (e.g. black, white, etc.).</li><li>• Unknown – this status is displayed during the first 30 seconds of the content analysis.</li></ul>

## P1/P2 Errors

The P1/P2 errors panel displays data about Priority 1 and Priority 2 errors that were found in the stream during the analysis. The table displays the following information for each error:

- **Indicator** – the indicator for which the error was generated.
- **#Errors** – the total number of errors that were generated.
- **Last error** – the time in which the last error was generated.
- **Error message** – the message that was generated with the error.

The statistics regarding the following Priority 1 and Priority 2 errors are displayed.

Parameter	Description
<b>Priority 1</b>	
<b>TS_sync_loss</b>	Indicates a loss of synchronization with the TS. When 5 sync bytes have been acquired, the decoder is considered synched, but a loss of just 2 sync bytes indicates a loss of sync.

## Broadcaster Inputs

### Viewing TR101 Analysis

<b>Sync_byte_error</b>	If the sync byte is not equal to 47 hexidecimals, then a sync byte error occurs. The system then looks for the reoccurrence of the sync byte (which must be 47 hexadecimals) every 188 bytes.
<b>PAT_error</b>	The program association table (PAT) is the only packet with packet ID (PID) Hex 0000, and it must occur at least every 0.5s to keep this error from occurring. Every program within the TS is listed in the PAT; if it is missing, then no programs can be decoded
<b>Continuity_count_error</b>	This error occurs when any of the following faults happen — incorrect packet order, a packet occurs more than twice or a packet is lost.
<b>PMT_error</b>	This error can occur if the program map table (PMT) does not come up at least every 0.5s on the PID that is referred to in the PAT.
<b>PID_error</b>	When TSes are remultiplexed, this can occur if any PID doesn't refer to an actual data stream.
<b>Priority 2</b>	
<b>Transport_error</b>	This flag is set in the TS header by the demodulator if it can't correct errors in the stream.
<b>CRC_error</b>	This indicates that a CRC error (data corruption) occurred in any of the following tables — CAT, PAT, PMT, NIT, EIT, BAT, SDT or TOT.
<b>PCR_error</b>	This flag is raised if the primary clock reference (PCR) is not seen for more than 100ms. The time interval between two consecutive PCR values should be no more than 40ms. This type of error can cause the decoder to lose lock on the 27MHz clock.
<b>PCR_accuracy_error</b>	This error can occur when the PCR accuracy of the selected program is outside the range of $\pm 500$ ns.
<b>PTS_error</b>	This occurs when the presentation time stamp (PTS) repetition is more than 700ms. The PTS is contained in the MPEG-2 program stream and is used to aid the decoder in presenting the program on time, at the correct speed and synchronized. The PTS is compared to the PCR.
<b>CAT_error</b>	This is used for conditional access programs (paid programming).

## Content Analysis

The Content Analysis errors panel displays that were found in the stream as part of Zixi's content analysis. To see this table, you must enable the content analysis, by following the instructions in the [Enabling Content Analysis](#) section. The table displays the following information for each error:

- **Indicator** – the indicator for which the error was generated.

## Broadcaster Inputs

---

### Viewing TR101 Analysis

- **#Errors** – the total number of errors that were generated.
- **Last error** – the time in which the last error was generated.
- **Error message** – the message that was generated with the error.

Parameter	Description
<b>Frozen_video</b>	Occurrence of frozen video, i.e. no motion, for a duration of approximately 60s or longer.
<b>Blank_picture</b>	Occurrence of blank frames of any color for a duration of approximately 60s or longer.
<b>Silent_audio</b>	Occurrence of silence (dBFS < -90) on AAC or MPEG audio with a duration of about 30s or longer.
<b>Low_video_quality</b>	The estimated video quality, which is based on the Peak signal-to-noise ratio (PSNR) is below the 20dB threshold and therefore considered as poor.
<b>Audio_clipping</b>	Occurrence of clipped audio upon occurrence on AAC or MPEG audio.

## Enabling Content Analysis

Zixi's content analysis can be enabled in two of the following modes:

- **Content Analysis: I-Frames** – The content analysis is performed on I-frames only. This option requires far less processing resources.
- **Content Analysis: All Frames** - The content analysis is performed on all frames. This may require more processing resources.

### ➔ To enable content analysis:

1. Add TR101 Analysis to an input stream. See [Adding TR101 Analysis to the Input Stream](#)
2. Click the stethoscope icon shown in the TR 101 290 column. The Stream Analysis screen is displayed.
3. On the Stream Analysis screen, select one of the following content analysis options from the Content Analysis drop-down menu:
  - **Content Analysis: I-Frames**
  - **Content Analysis: All Frames**

# Broadcaster Outputs

Zixi Broadcaster supports the following output types:

- **UDP** – Target is a UDP or RTP stream, over unicast or multicast.
- **RTMP** – Real Time Messaging Protocol streams can be played on systems that accept this protocol as input. Originally designed to enable Adobe® Flash® streams to be sent between a server and a client, RTMP has become a popular protocol for streaming into content-delivery networks (CDNs).
- **RTMPS** – RTMP over a TLS/SSL connection.
- **Pull Stream** – Pull streams are Zixi protocol streams destined for Zixi Receivers. Zixi Broadcaster can offer status and statistics for a stream that is being pulled from it by another device. **You normally do not need to set up a Pull output stream in Zixi Broadcaster.** If your application uses Zixi Receiver, you should configure the stream as input on Zixi Receiver and point it to the IP address and output port (default 2077) of the Zixi Broadcaster. Zixi Receiver will pull the stream and automatically create the proper output definition in the Zixi Broadcaster.
- **Push Stream** – Push streams are Zixi protocol streams destined for other Zixi Broadcasters and “Accept mode” Zixi Receivers.
- **TCP** - TCP protocol provides reliable transmission guarantee for no loss of frame. Its bandwidth probing and congestion control will attempt to use all of the available bandwidth between the server and client, fetching content as quickly as possible while being friendly to other (TCP) traffic on the same links. Output streams can be pushed to one Zixi Broadcasters or Zixi Receiver.
- **HTTP Push** - The HTTP Push output stream can be encapsulated in HLS or DASH. The source of the output stream is a predefined adaptive group that has been configured in the **Adaptive Groups** screen. The configuration of the stream includes the option to delete outdated stream fragments on the server. Output streams can be pushed to one Zixi Broadcasters or Zixi Receiver.
- **S3 Bucket** - The HTTP output stream (HLS or DASH) can be pushed to an AWS S3 Bucket. The native S3 support includes the ability to configure the S3 authentication credential, including the Region, Access Key, and Secret Key.
- **RIST** - The RIST output can be pushed or have them pulled to/from a RIST-enabled device. RIST seeks to provide reliable, high performance media transport by using UDP at the transport layer to avoid the inefficiencies of TCP, then layering on top Forward Error Correction and Retransmits to add the reliability inherently absent from UDP.
- **SRT** – The SRT output can be pulled to/from an SRT-enabled device and is configured via Zen Master or Zixi API only. SRT stands for ‘Secure Reliable Transport’ and is an open-source video transport protocol and technology stack. SRT uses secure streams and easy firewall traversal to optimize streaming performance and deliver high-quality video over even the most unreliable networks.

## Broadcaster Outputs

---

### Adding Outputs

When Zixi Receiver pulls a stream, a Pull stream definition is generated automatically in Zixi Broadcaster, the stream is identified in the Name and Destination columns of Zixi Broadcaster with the ID and IP address of the Zixi Receiver that pulls the stream.

#### Automated Output Configuration with Output Templates

Instead of manually creating an output and then connecting the output to an input, you can define output templates with rules for automatic creation of outputs with predefined settings. The template includes an Input pattern. When an input pattern matches the pattern of an incoming input stream, the output is created automatically. For more information about Output Templates, see Using Output Templates.

## Adding Outputs

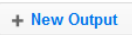
Zixi Broadcaster enables you to add various types of outputs. When you create a new output, select the output type and then fill in the fields that are applicable for that output type.

### Adding UDP Outputs

This enables the Broadcaster to send a UDP stream. The UDP stream is not protected by the Zixi protocol. The UDP streams can be one of the following types:

- UDP – plain UDP
- RTP – Real-Time Transport Protocol
- RTP + SMPTE 2022 FEC - SMPTE 2022 standard with Forward Error Correction

#### ➔ To Create a UDP Output:

4. In the Zixi Broadcaster **Outputs** page, click . The **New output** window is displayed.



## Broadcaster Outputs

### Adding Outputs

**New output**

Output Name:

Input Stream:

Show in Matrix:

UDP  RTMP  Pull  Push  TCP  HTTP Push  AWS  RIST

**Output parameters**

Host:

Port:  TTL:

Smoothing [ms]:

RTP:

Bind to IP:

Local port:

ReMuxing

Enable decryption

Enable IFB/Talkback

Ok Cancel

5. Type in the unique identifier for this output in the **Output Name**.
6. Select an **Input Stream** to be made available on this output from the drop-down list.
7. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
8. Select **UDP**.
9. Specify the **Output parameters** (see Output Stream Parameters Table).
10. In the **Host** field, specify the IP address or URL of the stream destination.
11. In the **Port** field, specify the port of the destination.
12. If you want to remux the stream to CBR (used for professional IRDs) in order to avoid bandwidth waste, select **Remuxing** and configure the respective parameters (see Output Stream Parameters Table).
13. Click **OK**.


The new UDP Output with its settings is created and now appears on the **Outputs** screen.

### Output Stream Parameters Table

Parameter	Description
-----------	-------------

## Broadcaster Outputs

### Adding Outputs

<b>Output Parameters</b>	
<b>Host</b>	Specify the destination IP address or URL.
<b>Port</b>	Specify the target UDP port.
<b>TTL</b>	Specify the time-to-live (TTL). Used mainly to limit the range (or “scope”) of a multicast transmission. Leaving it as blank uses the default value specified by the operating system.
<b>Smoothing [mms]</b>	Enables transmission of the output at the correct rate. Required when the receiving device is sensitive and can’t lock on to the stream (in such a case, the recommended latency is 100 - 1000 ms).
<b>RTP</b>	Enables use of the Real-time Transport Protocol.
	Disables the FEC or specify the type of FEC matrix to D1/ D2.
	<div data-bbox="402 877 457 949"></div> <p>This setting can be modified without interrupting an existing stream.</p>
<b>SMPTTE 2022 FEC:</b>	<p>In the SMPTE-2022 FEC standard, 1-D and 2-D parity codes are systematic FEC codes of decent complexity that provide protection against bursty or random losses. To enable SMPTE-2022 FEC (Forward Error Correction) select one of the following parity options from the drop-down menu:</p> <ul style="list-style-type: none"><li>• <b>1D</b> (single dimension FEC) - SMPTE-2022 1D can fix a problem in ONE packet per row. This means that the number of rows specified should consider the packet loss rate- the shorter the row, the more error resilience can be achieved on the expense of bitrate level. However, longer rows increase the delay that the FEC will add. It is recommended to select this option when the packet loss is sporadic. The stream will occupy the source port and two ports above it (e.g. if the source port is 8988, it will occupy 8990).</li><li>• <b>2D</b> - SMPTE-2022 2D uses a table of packets, which includes rows and columns. This option is more error resilient, but it increases the delay by (row*column) packets and increases the bitrate with R+C packets per each table of R*C packets. It is recommended to select this option when packet loss is bursty. The stream it will occupy four ports above (e.g. if the source port is 8988, it will occupy 8992).</li></ul>
<b>Bind to IP</b>	Select from the drop-down list the local IP address to be used for this Output, OR select <i>Any</i> to enable any IP to be used.
<b>Local Port</b>	Specify from which local port to source the stream, if desired (instead of randomly chosen by the operating system).

## Broadcaster Outputs

---

### Adding Outputs

<b>ReMixing</b>	Select this checkbox to enable remuxing from VBR MPEG-TS to MPEG-TS CBR (used for professional IRDs).
<b>Remux to bitrate [kbps]:</b>	Specify the CBR bitrate to target for a remuxed stream.
<b>Buffer Size [ms]</b>	Specify the size of the buffer that will be used for the remixing. A larger buffer increases the quality, but also increases latency.
<b>No NULL Packets</b>	Select this option to re-multiplex the stream to CBR but remove the NULL packets in order to maintain low bitrate.
<b>Enable Decryption</b>	Select this checkbox to decrypt an encrypted stream.
<b>Decryption type</b>	<p>If the stream has been encrypted, you can decrypt through the output by selecting this checkbox and specifying the type of encryption that was used.</p> <p>Select <b>Automatic</b> to decrypt a stream that has been encrypted using <b>Automatic</b> encryption.</p> <p>Only if the stream has not been encrypted, select <b>None</b> (default).</p> <p><b>Important Notice: if the stream has been encrypted you must decrypt it in the output.</b></p>
<b>Decryption key</b>	If the decryption type includes a key, enter the key to decrypt the stream during output.
<b>Enable IFB/Talkback</b>	Select this checkbox to enable IFB/Talkback for one-way communication from the director or assistant director to on-air talent or a remote location. Selecting the checkbox allows the decoder with a Zixi Receiver to send audio back to the encoder with a Zixi Feeder. This offers the ability to, for example, give audio feedback to the camera operator.
<b>Port</b>	The port that will be used to send the audio feedback.

## Broadcaster Outputs

### Adding Outputs

## Adding RTMP and RTMPS Outputs

Real Time Messaging Protocol streams can be played on systems that accept this protocol as input. Originally designed to enable Adobe® Flash® streams to be sent between a server and a client, RTMP has become a popular protocol for streaming into content-delivery networks (CDNs).



To create an RTMPS output, select the RTMP output option but make sure `rtmps://` url is used in the URL field.



SCTE-35 markers in MPEG-TS streams trigger OnCuePoint call function in RTMP.

### ➔ To create an RTMP Output:

1. In the Zixi Broadcaster **Outputs** page, click [+ New Output](#). The **New output** window is displayed.

**New output**

Output Name:

Input Stream:

Show in Matrix:

UDP  RTMP  Pull  Push  TCP  HTTP Push  AWS  RIST

**Output parameters**

URL (rtmp://host:[port]/app/stream):   
rtmp://localhost/live/input

Backup URL:

Stream name:

Username:

Password:

Bitrate [kbps]:

Reconnect [sec]:

Send timecode:

Disconnect if inactive:

Ignore TLS certificate:

Enable decryption

Ok Cancel

2. Type the unique identifier for this output in the **Output Name**.

## Broadcaster Outputs

---

### Adding Outputs

3. Select an **Input Stream** to be made available on this output from the drop-down list.
4. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
5. Select **RTMP**.
6. Specify the **Output parameters** (see Output Stream Parameters Table).
7. Click **OK**.
8. The new RTMP Output with its settings is created and now appears on the screen.

### Output Stream Parameters Table

Parameter	Description
<b>Output Parameters</b>	
<b>URL</b>	Specify the destination Stream URL according to the following format: <code>rtmp://host:[port]/app</code> . You can define RTMPS clients with the proven security of a secure socket layer (SSL), by using the <code>rtmps://</code> url. Note that Facebook currently supports RTMPS streams only.
<b>Backup URL</b>	Enter a backup URL (fallback mode) – will be used in case the primary server is not responsive.
<b>Stream Name</b>	Specify the unique name for the stream that the RTMP server at the other end of the connection expects.
<b>Username</b>	Specify the Username to authenticate on the remote RTMP server
<b>Password</b>	Specify the string that is used for authorization on the remote RTMP server.
<b>Bitrate (kbps)</b>	Specify the actual bitrate in kbps or the maximum stream bitrate in case of a VBR stream.
<b>Reconnect (sec)</b>	If the stream drops, this value represents the time between reconnection attempts (in seconds). Default: 5.
<b>Send Timecode</b>	Converts the MPEG-TS SEI section to RTMP ONFI command (pass the encoder timecodes to the RTMP server)
<b>Disconnect if inactive</b>	Select this checkbox to disconnect if the stream is not active for 30 seconds.
<b>Ignore TLS certificate errors</b>	Select this option when streaming to a trusted server that does not have a valid certificate. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.

## Broadcaster Outputs

### Adding Outputs

<b>Enable Decryption</b>	Select this checkbox to enable decryption of an encrypted stream.
<b>Decryption type</b>	<p>If the stream has been encrypted, you can decrypt through the output by selecting this checkbox and specifying the type of encryption that was used.</p> <p>Select <b>Automatic</b> to decrypt a stream that has been encrypted using <b>Automatic</b> encryption.</p> <p>Only if the stream has not been encrypted, select <b>None</b> (default).</p> <p><b>Important Notice: if the stream has been encrypted you must decrypt it in the output.</b></p>

### Adding Pull Outputs

#### ➔ To Create a Pull Output:

1. In the Zixi Broadcaster **Outputs** page click [+ New Output](#). The **New Output** window is displayed.

**New output**

Output Name:

Input Stream:

Show in Matrix:

UDP  RTMP  Pull  Push  TCP  HTTP Push  AWS  RIST

**Output parameters**

Stream ID:

Remote ID:

Password:

Latency [ms]:  Remote configuration

Enable encryption

Enable IFB/Talkback

Ok Cancel

2. Type in the unique identifier for this output in the **Output Name**.
3. Select an **Input Stream** to be made available on this output from the adjacent drop-down list.
4. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
5. Select **Pull**.

## Broadcaster Outputs

---

### Adding Outputs

6. Enter the **Output Parameters** (see Output Stream Parameters Table).
7. Click **OK**.

The new Pull Output with its settings is created and now appears on the screen.

### Output Stream Parameters Table

Parameter	Description
<b>Output Parameters</b>	
<b>Stream ID</b>	Specify the unique Stream ID that the remote Zixi Receiver/Broadcaster expects (recommended – identical to 'Input Stream').
<b>Remote ID</b>	Specify the Zixi Receiver/Broadcaster ID (shown in the Zixi Receiver/Broadcaster <b>Status</b> window) that will be allowed to pull the stream.
<b>Password</b>	Specify a string that can be used to authenticate that the Zixi Receiver is authorized to pull this stream (Optional).
<b>Latency [ms]</b>	Specify the buffer size that Zixi Broadcaster will be able to maintain for error correction. For example, 6000 milliseconds would ensure that the stream is protected for six seconds of errors in the network. Default: Remote configuration. Remote configuration – use the latency configured in the Zixi Receiver/Broadcaster. Otherwise use a value to override the value configured in the Zixi Receiver/Broadcaster.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Output stream.
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256). Click <b>Generate</b> to generate an encryption key or enter another encryption key.
<b>Enable IFB/Talkback</b>	Select this checkbox to enable IFB/Talkback for one-way communication from the director or assistant director to on-air talent or a remote location. Selecting the checkbox allows the decoder with a Zixi Receiver to send audio back to the encoder with a Zixi Feeder. This offers the ability to, for example, give audio feedback to the camera operator.
<b>Port</b>	The port that will be used to send the audio feedback.

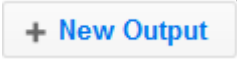
## Broadcaster Outputs

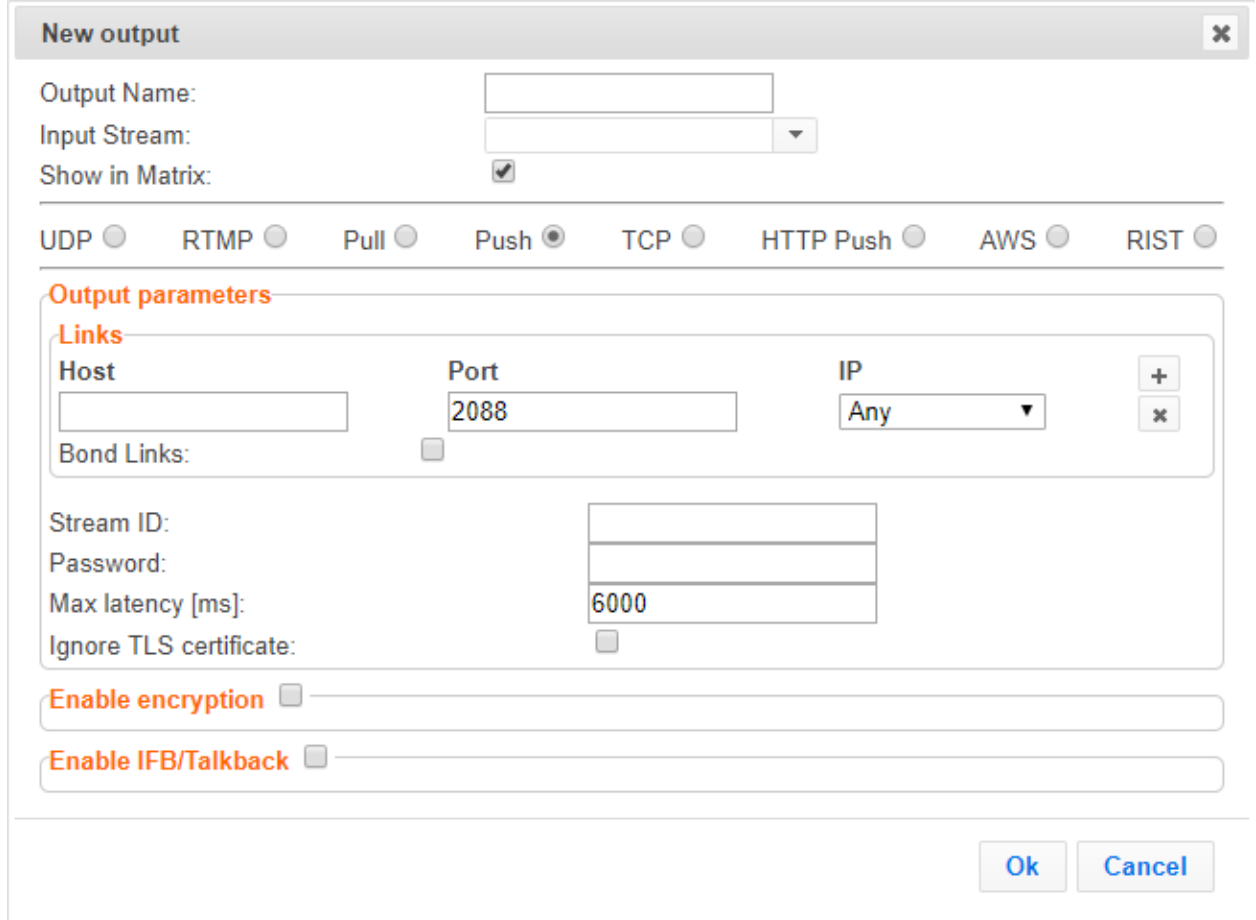
### Adding Outputs

## Adding Push Outputs

Output streams can be pushed to one or more Zixi Broadcasters and Zixi Receivers. You can specify multiple destinations (Broadcasters/Receivers) by adding them in the Links section. You can also split a single stream using multiple links and then bond them into a single stream in target Zixi Broadcaster using the Bond Links feature. For each of these links, you can specify the max bitrate that can be used. In addition, some links may be specified as backups. The backup links will be used if the bandwidth across all the non-backup links are not enough for the stream (i.e. if a non-backup link is not available, it will not use the backup). Only if the accumulated bandwidth of non-backup links is not enough, then the Broadcaster will stream to a backup link.

### ➔ To Create a Push Output:

1. In the Zixi Broadcaster **Outputs** page, click . The **New Output** window is displayed.



2. Type in the unique identifier for this output in the **Output Name**.
3. Select an **Input Stream** to be made available on this output from the adjacent drop-down list.
4. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.



## Broadcaster Outputs

---

### Adding Outputs

5. Select **Push**.
6. Do one of the following:
  7. **Stream to multiple destinations** - you can stream the same identical stream to multiple destinations by entering the destination parameters and then clicking the "+" button to add another destination (see **Output Stream Parameters** Table).
  8. **Bonded link to a single destination** - you can split the stream into multiple links, which will be forwarded to a single Zixi Broadcaster by selecting the **Bond Links** checkbox (see **Using Network Bonding for Hitless Failover**).
9. Enter additional parameters as necessary (see Output Stream Parameters Table)
10. Click **OK**. The new Push Output with its settings is created and now appears on the screen.

### Output Stream Parameters Table

Parameter	Description
<b>Output Parameters</b>	
<b>Links</b>	
<b>Host</b>	The destination host for the push stream. <b>Note:</b> additional failover destinations can be added by clicking the adjacent "+" button.
<b>Port</b>	Specify the ports to listen on for Push streams. Default: 2088.
<b>IP</b>	Specify the local IP address.
<b>Max Bitrate (for Bond Links enabled)</b>	Relevant for <b>Bond Links</b> option. Specify the max bitrate in Kbps for the specific link.
<b>Bond Links</b>	Click this checkbox to split the stream into multiple links, which will be forwarded to a single Zixi Broadcaster.
<b>Backup (for Bond Links enabled)</b>	Relevant for Bond Links option. Select this checkbox to define the link as a backup. The backup will be used only if the accumulated bandwidth of the non-backup links is not sufficient.
<b>Stream ID</b>	Specify the unique Stream ID that the remote Zixi Broadcaster expects.
<b>Password</b>	Optional - Specify a string to be used at the receiving Zixi Broadcaster to authenticate that the pushing Broadcaster is allowed to push this stream.

## Broadcaster Outputs

---

### Adding Outputs

---

<b>Max Latency [ms]</b>	Specify the buffer size that Zixi Broadcaster will keep for error correction. For example, 6000 milliseconds would ensure that the stream is protected for six seconds of errors in the network. Default:6000
<b>Ignore TLS certificate errors</b>	Select this option when streaming to a trusted server that does not have a valid certificate. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Output stream.
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256). Click <b>Generate</b> to generate an encryption key or enter another encryption key.
<b>Enable IFB/Talkback</b>	Select this checkbox to enable IFB/Talkback for one-way communication from the director or assistant director to on-air talent or a remote location. Selecting the checkbox allows the decoder with a Zixi Receiver to send audio back to the encoder with a Zixi Feeder. This offers the ability to, for example, give audio feedback to the camera operator.
<b>Port</b>	The port that will be used to send the audio feedback.

---

## Adding TCP Outputs

TCP protocol provides reliable transmission guarantee for no loss of frame. Its bandwidth probing and congestion control will attempt to use all of the available bandwidth between the server and client, fetching content as quickly as possible while being friendly to other (TCP) traffic on the same links. The TCP output can push the stream directly to a TCP socket or to an HTTP server.

### MultiPath TCP

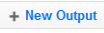
Developed by the Internet Engineering Task Force's (IETF) Multipath TCP working group, MultiPath TCP (MPTCP) is an effort towards enabling the simultaneous use of several IP-addresses/interfaces by a modification of TCP that presents a regular TCP interface to applications, while in fact spreading data across several subflows. Zixi supports MultiPath TCP for Linux based on the OS Kernel support. This means that the Linux Kernel should be updated to support MPTCP. For detailed information on how to update the Kernel and use MPTCP, see [Configuring MultiPath TCP](#).

The MPTCP functionality is similar to Zixi's "multi-link bonding" feature, which is available for UDP only. In this case MPTCP works on TCP.

If both the sending end and the receiving end support MultiPath TCP, Zixi can use this functionality to transmit a stream over TCP using multiple network interfaces (e.g. Wifi + mobile). The multiple links can be configured to:

- **Aggregate** - aggregate the bandwidth of the multiple links in order to use the total capacity of the links. In this case Zixi will use all available bandwidth among all available flows (if needed) to send the stream.
- **Duplicate** - enhance the path's resiliency by duplicating the stream and sending through multiple links in order to overcome errors and packet loss.

### ➔ To Create a TCP Output:

1. In the Zixi Broadcaster **Outputs** page, click . The **New Output** window is displayed.

## Broadcaster Outputs

### Adding Outputs

The screenshot shows a 'New output' dialog box with the following fields and options:

- Output Name: [Text input]
- Input Stream: [Dropdown menu]
- Show in Matrix:
- Protocol selection: UDP, RTMP, Pull, Push, TCP (selected), HTTP Push, AWS, SRT
- Output parameters section:
  - Host: [Text input]
  - Port: [Text input]
  - HTTP:
  - Multi Path TCP:
  - Mode: [Dropdown menu with 'Aggregate' selected]
  - Enable decryption: [Text input with 'Duplicate' option]
- Buttons: Ok, Cancel

2. Type in the unique identifier for this output in the **Output Name**.
3. Select an **Input Stream** to be made available on this output from the adjacent drop-down list.
4. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
5. Select **TCP**.
6. Specify the Output parameters (see **Output Stream Parameters** Table).
7. If the target is an HTTP server, select the **HTTP** checkbox. In this case, HTTP headers will be added to the stream.
8. Enter additional parameters as necessary (see Output Stream Parameters Table).
9. Click **OK**. The new TCP Output with its settings is created and now appears on the screen.

### Output Stream Parameters Table

Parameter	Description
<b>Output Parameters</b>	
<b>Host</b>	The destination host for the stream.
<b>Port</b>	Specify the ports to listen on for TCP streams. Default: 7777.
<b>HTTP</b>	If the target is an HTTP server, select the <b>HTTP</b> checkbox. In this case, HTTP headers will be added to the stream.

## Broadcaster Outputs

---

### Adding Outputs

<b>Multi Path TCP</b>	Select this checkbox to enable MultiPath TCP, which enables the transmission of a stream over TCP using two network interfaces (e.g. Wifi + mobile). Multi Path TCP is available for Linux only.
<b>Mode</b>	<p>The multiple links can be configured to:</p> <ul style="list-style-type: none"><li>• <b>Aggregate</b> - aggregate the bandwidth of the multiple links in order to use the total capacity of both links. In this case Zixi will use all available bandwidth among all available flows (if needed) to send the stream.</li><li>• <b>Duplicate</b> - enhance the path's resiliency by duplicating the stream and sending through multiple links in order to overcome errors and packet loss.</li></ul>
<b>Enable Decryption</b>	Select this checkbox to enable decryption of an encrypted stream.
<b>Decryption type</b>	<p>If the stream has been encrypted, you can decrypt through the output by selecting this checkbox and specifying the type of encryption that was used.</p> <p>Select <b>Automatic</b> to decrypt a stream that has been encrypted using Automatic encryption.</p> <p>Only if the stream has not been encrypted, select <b>None</b> (default).</p> <p>Important Notice: if the stream has been encrypted you must decrypt it in the output.</p>
<b>Decryption key</b>	Enter the decryption key to decrypt the encrypted stream.

## Adding HTTP Push Outputs

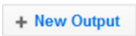
The HTTP Push output stream can be encapsulated in HLS or DASH. The source of the output stream is a predefined adaptive group that has been configured in the **Adaptive Groups** screen. The configuration of the stream includes the option to delete outdated stream fragments on the server. SCTE-35 markers in the source input will be preserved and used to segment the stream for both HLS and DASH encapsulation. Output streams can be pushed to one Zixi Broadcaster or Zixi Receiver.

The path to the destination can be formed in one of the following ways:

- **Full path** – in this option the path would be constructed as follows: `<base url>/<path>/<file>?<additional params may be also with path and file>`
- **Path in parameters** - in this option the path would be constructed as follows: `<base url>?<additional params may be also with path and file>`
- **Flat path** - in this option the path would be constructed as follows: `<base url>/<path>_<file>?<additional params may be also with path and file>`. Any "/" in path should be replaced with "\_"

**Low Latency HLS** - Zixi Broadcaster supports low latency HLS delivery using chunked transfer encoding, which starts sending an HTTP response as chunks of data when they are ready before the complete response is available. Chunked transfer encoding can be used with transport stream (TS) segments and with fragmented MP4 (fMP4). For low latency HLS and DASH, Zixi Broadcaster must be used as an origin server. To enable low latency HLS and DASH, select **Enable HLS Fast playback** on **Settings > Live Protocols** screen.

### ➔ To Create an HTTP Push Output:

1. In the Zixi Broadcaster **Outputs** page, click . The **New Output** window is displayed.

## Broadcaster Outputs

### Adding Outputs

**New output**

Output Name:

Input Adaptive Group:

Show in Matrix:

UDP  RTMP  Pull  Push  TCP  HTTP Push  AWS  RIST

**Output parameters**

HLS encapsulation  DASH/fMP4 HLS encapsulation

Target URL:

Additional URL parameters:

URL parameters support macros: %PATH% - file path as appears in playlist; %FILE% - file name

Delete outdated fragments from server

Upload DVR playlist

Segment Path:

Ignore TLS certificate errors

Authorization

Ok Cancel

2. Select the **HTTP Push** option (radio button).
3. Type in the unique identifier for this output stream in the **Output Name**.
4. Select an **Input Adaptive Group** to be made available on this output from the adjacent drop-down list. To create an Adaptive Group, see [Adaptive Groups](#).
5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: not selected.
6. Enter the **Output Parameters** (see Output Stream Parameters Table).
7. Click **OK**.

### Output Stream Parameters Table

Parameter	Description
<b>Output Parameters</b>	

## Broadcaster Outputs

---

### Adding Outputs

---

---

---

**Encapsulation Options**

- **HLS Encapsulation** - the output stream, which includes all the streams in the selected Adaptive group, will be encapsulated in HTTP Live Streaming format.
  - **Dash Encapsulation** - the output stream, which includes all the streams in the selected Adaptive group, will be encapsulated in MPEG DASH format.
- 

Depending on the selection in the **Segment Path** field below, this field should include the following:

**Target URL**

- **Full path** - `<base url>/<path>/<file>`
  - **Path in parameters** - in this option the path would be constructed as follows: `<base url>`
  - **Flat path** - in this option the path would be constructed as follows: `<base url>/<path>_<file>`
- 

Depending on the selection in the **Segment Path** field below, this field should include the following:

**Additional URL parameters**

- **Full path** - `?<additional params may be also with path and file>`
- **Path in parameters** - `?<additional params may be also with path and file>`
- **Flat path** - `?<additional params may be also with path and file>`. Any "/" in path should be replaced with "\_"

URL parameters support macros as follows:

- `%PATH%` - file path as it appears in the playlist
  - `%FILE%` - file name
- 

**Delete outdated stream fragments on server**

Select this checkbox to delete any outdated stream fragments from the server.

---

**Upload DVR playlist**

Select this checkbox if you are streaming a recorded/VOD stream and you want to upload a DVR playlist in addition to the live playlist.

---



## Broadcaster Outputs

### Adding Outputs

---

---

Select one of the following path segmentation types:

<b>Segment Path</b>	<ul style="list-style-type: none"><li>• <b>Full path</b> – in this option the path would be constructed as follows: <code>&lt;base url&gt;/&lt;path&gt;/&lt;file&gt;?&lt;additional params may be also with path and file&gt;</code></li><li>• <b>Path in parameters</b> - in this option the path would be constructed as follows: <code>&lt;base url&gt;?&lt;additional params may be also with path and file&gt;</code></li><li>• <b>Flat path</b> - in this option the path would be constructed as follows: <code>&lt;base url&gt;/&lt;path&gt;_&lt;file&gt;?&lt;additional params may be also with path and file&gt;</code>. Any "/" in path should be replaced with "_"</li></ul>
<b>Ignore TLS certificate errors</b>	Select this option when streaming to a trusted server that does not have a valid certificate. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Authorization</b>	Select this checkbox if your destination server requires authorization credentials to connect.

Select one of the following authorization methods:

<b>Method</b>	<ul style="list-style-type: none"><li>• Basic authorization</li><li>• Digest authorization</li><li>• OAuth2 authorization</li></ul>
<b>Username</b>	Type the username
<b>Password</b>	Type the password

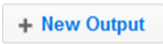
---

---

### Adding AWS Outputs

The HTTP output stream, which is encapsulated in HLS or DASH/fMP4 HLS can be pushed to either an AWS S3 Bucket or to an AWS Elemental MediaStore. The native AWS support includes the ability to configure the S3 authentication credential, including the Region, Access Key, and Secret Key.

#### ➔ To Create an AWS Output:

1. In the Zixi Broadcaster **Outputs** page, click . The **New Output** window is displayed.

## Broadcaster Outputs

### Adding Outputs

**New output**

Output Name:

Input Adaptive Group:

Show in Matrix:

UDP  RTMP  Pull  Push  TCP  HTTP Push  AWS  RIST

**Output parameters**

Service: S3 Bucket  Elemental MediaStore

Encapsulation: HLS  DASH/MP4 HLS

URL:

Delete outdated stream fragments on server:

Ignore TLS certificate errors

\*TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot(s)

Access key:

Secret key:

Segment Path:

Ok Cancel

2. Select the **AWS** option (radio button).
3. Type in the unique identifier for this output stream in the **Output Name**.
4. Select an **Input Adaptive Group** to be made available on this output from the adjacent drop-down list. To create an Adaptive Group, see [Adaptive Groups](#).
5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: not selected.
6. Enter the **Output Parameters** (see Output Stream Parameters Table).
7. Click **OK**.

### Output Stream Parameters Table

Parameter	Description
<b>Output Parameters</b>	
<b>Service</b>	Select one of the following options: <ul style="list-style-type: none"><li>• S3 Bucket - the output stream will be pushed to the specified S3 Bucket.</li><li>• Elemental Media Store - the output stream will be pushed to the specified AWS Elemental MediaStore.</li></ul>
<b>Encapsulation</b>	<ul style="list-style-type: none"><li>• HLS Encapsulation - the output stream, which includes all the streams in the selected Adaptive group, will be encapsulated in HTTP Live Streaming format.</li><li>• Dash Encapsulation - the output stream, which includes all the streams in the selected Adaptive group, will be encapsulated in MPEG DASH format.</li></ul>

## Broadcaster Outputs

---

### Adding Outputs

---

---

<b>URL</b>	<p>The URL address of the destination host for the S3 Bucket or the AWS Elemental MediaStore.</p> <p>The path to the destination can be formed in one of the following ways according to the option selected in the <b>Segment Path</b> field below:</p> <ul style="list-style-type: none"><li>• <b>Full path</b> – in this option the path would be constructed as follows: &lt;base url&gt;/&lt;path&gt;/&lt;file&gt;?&lt;additional params may be also with path and file&gt;</li><li>• <b>Path in parameters</b> - in this option the path would be constructed as follows: &lt;base url&gt;?&lt;additional params may be also with path and file&gt;</li><li>• <b>Flat path</b> - in this option the path would be constructed as follows: &lt;base url&gt;/&lt;path&gt;_&lt;file&gt;?&lt;additional params may be also with path and file&gt;.</li></ul> <p>Any "/" in path should be replaced with "_"</p>
<b>Delete outdated stream fragments on server</b>	Select this checkbox to send delete messages to S3/AWS Elemental MediaStore to delete the outdated stream fragments from the servers.
<b>Ignore TLS certificate errors</b>	Select this option when streaming to a trusted server that does not have a valid certificate. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access key</b>	Enter your AWS S3 or AWS Elemental MediaStore Access key.
<b>Secret key</b>	Enter your AWS S3 or AWS Elemental MediaStore Secret key.
<b>Segment Path</b>	<p>Select one of the following path segmentation types:</p> <ul style="list-style-type: none"><li>• <b>Full path</b> – in this option the path would be constructed as follows: &lt;base url&gt;/&lt;path&gt;/&lt;file&gt;?&lt;additional params may be also with path and file&gt;</li><li>• <b>Path in parameters</b> - in this option the path would be constructed as follows: &lt;base url&gt;?&lt;additional params may be also with path and file&gt;</li><li>• <b>Flat path</b> - in this option the path would be constructed as follows: &lt;base url&gt;/&lt;path&gt;_&lt;file&gt;?&lt;additional params may be also with path and file&gt;.</li></ul> <p>Any "/" in path should be replaced with "_"</p>

### Adding RIST Outputs

The RIST output can be push streams or have them pulled to/from a RIST-enabled device. RIST seeks to provide reliable, high performance media transport by using UDP at the transport layer

## Broadcaster Outputs

---

### Adding Outputs

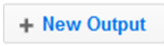
to avoid the inefficiencies of TCP, then layering on top Forward Error Correction and Retransmits to add the reliability inherently absent from UDP.

Zixi supports the following RIST profiles:

- **Simple Profile (PUSH only)** - The RIST Simple Profile provides best-in-class packet protection, as well as support for multi-link bonding and seamless switching.
- **Main Profile (PUSH/PULL)** – The RIST Main Profile builds upon the features of the Simple Profile to provide backwards compatibility, while opening the door for additional features, including:
  - **Encryption and authentication** - RIST Main Profile uses industry-standard DTLS or PSK with a variety of available cyphers to ensure the most up-to-date protection, while being flexible to accommodate local legal regulations. Support for pre-shared keys is also included, which makes RIST Main Profile compatible with one-to-many environments such as satellite or multicast distribution.
  - **Native tunneling and multiplexing** - the ability to combine multiple Simple Profile RIST streams into one single connection, optionally with one single encryption and authentication step, using industry-standard GRE over UDP.
  - **Bandwidth optimization** - by deleting NULL packets at the source and re-inserting them at the destination.
  - **Support for high bit rate streams** - The RIST Main Profile includes an RTP header extension which increases the sequence number size to 32-bits, and therefore increases the maximum size of retransmission buffers, and thus opens the door for higher bitrate streams, such as lightly compressed (e.g. JPEG 2000, Tico) and uncompressed video.

Contact your Zixi representative to learn more about interoperability testing between Zixi Broadcaster and 3rd party devices. More details about RIST can be found at [www.rist.tv](http://www.rist.tv)

#### ➔ To Add a RIST Output:

1. In the Zixi Broadcaster **Outputs** page, click . The **New Output** window is displayed.

## Broadcaster Outputs

### Adding Outputs

New output

Output Name:

Input Stream:

Show in Matrix:

UDP  RTMP  Pull  Push  TCP  HTTP Push  AWS  RIST

**Output parameters**

Profile:

Host:

Port:  TTL:

Smoothing [ms]:

SMPTE-2022 FEC:

Bind to IP:

Local port:

ReMuxing

Enable decryption

Enable IFB/Talkback

Ok Cancel

2. Select the **RIST** option (radio button).
3. Type in the unique identifier for this output stream in the **Output Name**.
4. Select an **Input Stream** to be made available on this output from the adjacent drop-down list.
5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: not selected.
6. Enter the **Output Parameters** (see Output Stream Parameters Table).
7. Click **OK**.


The new RIST Output with its settings is created and now appears on the Outputs screen.

### Output Stream Parameters Table

Parameter	Description
<b>Output Parameters</b>	
	Select one of the following RIST profiles that matches the input stream:
<b>Profile</b>	<ul style="list-style-type: none"><li>• Simple Profile (default)</li><li>• Main Profile</li></ul>
	Select the stream direction:
<b>Type (Main Profile)</b>	<ul style="list-style-type: none"><li>• Push</li><li>• Pull</li></ul>
<b>Host</b>	Specify the destination IP address or URL of the host to which the steam will be pushed.

## Broadcaster Outputs

### Adding Outputs

<b>Port</b>	Specify the target UDP port.
<b>TTL</b>	Specify the time-to-live (TTL). Used mainly to limit the range (or “scope”) of a multicast transmission. Leaving it as blank uses the default value specified by the operating system
<b>Smoothing [ms]</b>	Enables transmission of the output at the correct rate. Required when the receiving device is sensitive and can’t lock on to the stream (in such a case, the recommended latency is 100 - 1000 ms)
<b>SMPTE-2022 FEC</b>	<p>In the SMPTE-2022 FEC standard, 1-D and 2-D parity codes are systematic FEC codes of decent complexity that provide protection against bursty or random losses.</p> <p> This setting can be modified without interrupting an existing stream.</p> <p>To enable SMPTE-2022 FEC (Forward Error Correction) select one of the following parity options from the drop-down menu:</p> <ul style="list-style-type: none"><li>• <b>1D</b> - SMPTE-2022 1D can fix a problem in ONE packet per row. This means that the number of rows specified should consider the packet loss rate- the shorter the row, the more error resilience can be achieved on the expense of bitrate level. However, longer rows increase the delay that the FEC will add. It is recommended to select this option when the packet loss is sporadic. The stream will occupy the source port and two ports above it (e.g. if the source port is 8988, it will occupy 8990).</li><li>• <b>2D</b> - SMPTE-2022 2D uses a table of packets, which includes rows and columns. This option is more error resilient, but it increases the delay by (row*column) packets and increases the bitrate with R+C packets per each table of R*C packets. It is recommended to select this option when packet loss is bursty. The stream will occupy four ports above (e.g. if the source port is 8988, it will occupy 8992).</li></ul>
<b>Bind to IP</b>	Select from the drop-down list the local IP address to be used for this Input, OR select <i>Any</i> to enable any IP to be used.
<b>Local Port</b>	Specify from which local port to source the stream, if desired (instead of randomly chosen by the operating system).
<b>RIST main profile parameters</b>	
<b>Use DTLS</b>	

## Broadcaster Outputs

---

### Adding Outputs

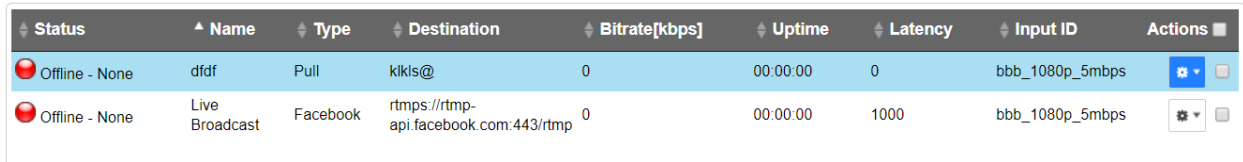
<b>ReMuxing</b>	Select this checkbox to enable remuxing from VBR MPEG-TS to MPEG-TS CBR (used for professional IRDs).
<b>Remux to bitrate [kbps]:</b>	Specify the CBR bitrate to target for a remuxed stream.
<b>Buffer Size [ms]</b>	Specify the size of the buffer that will be used for the remuxing. A larger buffer increases the quality, but also increases latency.
<b>No NULL Packets</b>	Select this option to re-multiplex the stream to CBR but remove the NULL packets in order to maintain low bitrate.
<b>Enable Decryption</b>	Select this checkbox to decrypt an encrypted stream.  If the stream has been encrypted, you can decrypt through the output by selecting this checkbox and specifying the type of encryption that was used. Select <b>Automatic</b> to decrypt a stream that has been encrypted using <b>Automatic</b> encryption. Only if the stream has not been encrypted, select <b>None</b> (default). <b>Important Notice: if the stream has been encrypted you must decrypt it in the output.</b>
<b>Decryption key</b>	If the decryption type includes a key, enter the key to decrypt the stream during output.
<b>Enable IFB/Talkback</b>	Select this checkbox to enable IFB/Talkback for one-way communication from the director or assistant director to on-air talent or a remote location. Selecting the checkbox allows the decoder with a Zixi Receiver to send audio back to the encoder with a Zixi Feeder. This offers the ability to, for example, give audio feedback to the camera operator.
<b>Port</b>	The port that will be used to send the audio feedback.

## Broadcaster Outputs

### Viewing Existing Outputs

## Viewing Existing Outputs

The Outputs screen displays all the output streams that have been created.



Status	Name	Type	Destination	Bitrate[kbps]	Uptime	Latency	Input ID	Actions
Offline - None	dfdf	Pull	kikls@	0	00:00:00	0	bbb_1080p_5mbps	
Offline - None	Live Broadcast	Facebook	rtmps://rtmp-api.facebook.com:443/rtmp	0	00:00:00	1000	bbb_1080p_5mbps	

Hovering the cursor on the status indicator of a stream shows the stream configuration information and basic utilization statistics, where applicable.

Any of the columns in the following table can be sorted in ascending/descending order by clicking on its heading. The **Search** text box in the upper left-hand side of the screen can be used to search for a specific input stream by entering the desired Stream ID.

Field	Description
<b>Status</b>	Displays the status of output stream. The following colored indicators are displayed: <ul style="list-style-type: none"><li>• Green = Connected</li><li>• Yellow = Connecting</li><li>• Red = Disconnected with an error condition</li><li>• Blue = Offline with no connection attempts</li><li>• Grey = Offline and stopped</li></ul>
<b>Name</b>	Displays the name of the output stream, as defined in the Zixi Broadcaster.
<b>Type</b>	Displays the type of output stream.
<b>Destination</b>	Displays the IP address of the host that will be receiving the stream.
<b>Bitrate [kbps]</b>	Displays bitrate of the connected output stream.
<b>Uptime</b>	Displays the amount of time the stream has been active.
<b>Latency</b>	Displays the configured latency
<b>Input ID</b>	The ID of the related Input stream.
<b>Actions</b>	Provides access to various actions to be performed on this output stream (see <a href="#">Performing Actions on Output Streams</a> below).




















## Broadcaster Outputs

### Viewing Existing Outputs

## Stats Pane

The Stats pane on the bottom of the page shows data about the activity of the selected stream.

Stats		
Bitrate (kbps)  : 0	Total Packets  : 0	FEC Packets  : 0
Last Connected: 0	Packet Rate  : 0	FEC Recovered  : 0
Send Errors  : 0	Packet Loss %  : 0	ARQ Requests  : 0
Reconnections: 0	Dropped Packets  : 0	ARQ Recovered  : 0
RTT(ms)  : 0	Recovered Packets  : 0	ARQ Duplicates  : 0
Jitter(ms)  : 0	Not Recovered Packets  : 0	Overflows  : 0

Click the  icon next to a parameter to show the graph of historical data for the specified parameter.

Field	Description
<b>Bitrate (kbps)</b>	Displays the current bitrate of the output stream
<b>Last Connected</b>	Displays the date and time in which the stream was connected.
<b>Send Errors</b>	Displays the number of errors generated after sending the stream.
<b>Reconnections</b>	Displays the number of reconnection attempts.
<b>RTT (ms)</b>	Displays the Round Trip Time (RTT) from the source (in milliseconds)
<b>Jitter</b>	Displays the current Jitter, measured in milliseconds.
<b>Total Packets</b>	Displays the total number of packets that have been transmitted.
<b>Packet Rate</b>	Displays the current number of packets per second.
<b>Packet Loss %</b>	Displays the packet loss rate (how many packets were lost out of the total number of packets).
<b>Dropped Packets</b>	Displays the total number of packets that were dropped since the beginning of the stream.
<b>Recovered Packets</b>	Displays the total number of dropped packets that have been recovered since the beginning of the stream.
<b>Not Recovered Packets</b>	Displays the number of non-recovered packets.
<b>Failed Sends</b>	Displays the number of failed attempts to send the stream.
<b>FEC Packets</b>	Displays the number of FEC packets transmitted.
<b>FEC Recovered</b>	Displays the number of FEC packets that were recovered.
<b>ARQ Requests</b>	Displays the number of requests for transmission of dropped packets made with ARQ.
<b>ARQ Recovered</b>	Displays the number of dropped packets recovered via ARQ.

## Broadcaster Outputs

---

### Viewing Existing Outputs

---

---


<b>ARQ Duplicates</b>	Displays the number of duplicate recovery packets received via ARQ.
<b>Overflows</b>	Displays the number of packets lost due to buffer overflow. Note: overflow is usually the result of the Max Bitrate being too low.

---

---

## Searching for Output Streams

### ➔ To Search for an Output Stream:

In the Zixi Broadcaster **Outputs** page, type in the specific output stream name in the search text box on the left-hand top of the screen and click . The **Output** screen displays the requested output stream(s).


## Performing Actions on Output Streams

Every stream has a shortcut button that can be used to perform relevant actions on the stream. The following actions can be performed through the Actions button:

- **Deleting Output Streams from Broadcaster**
- **Switching Input**


### Deleting Output Streams from Broadcaster

#### ➔ To Delete an Output Stream from the Broadcaster:

1. In the Zixi Broadcaster **Outputs** page, select the output stream you want to delete, and then click .
2. Click **Delete** to the selected stream. The selected input stream was deleted and is not displayed anymore.

### Switching Input

#### ➔ To change the input associated with the selected output:


1. In the Zixi Broadcaster **Outputs** page, click  and select **Switch Input**. The **Switch source** window is displayed.
2. In the **New Input** drop-down, select the desired input, and click **OK**. The newly designated input is now associated with the selected output.

## Viewing Traceroutes

You can view traceroute data for a stream sent from a Zixi Broadcaster (v13 or above) to a Zixi Receiver (v13 or above).

#### ➔ To view traceroute data:

1. On the **Outputs** page of Zixi Broadcaster, click the desired row of the output stream for which you want to view traceroutes.

Click the  button on the right end of the selected row and then select **Trace**.

## Broadcaster Outputs

---

### Performing Actions on Output Streams

The traceroute data for this stream is displayed.

## Performing Actions on Multiple Outputs

### Delete Marked Individual and Multiple Outgoing Stream

#### ➔ To Delete Output Streams:

1. In the Zixi Broadcaster Outputs page, select all outputs by selecting the checking the Actions checkbox at the header line, or select individual outputs by selecting their checkboxes.

The Marked drop-down button is displayed.



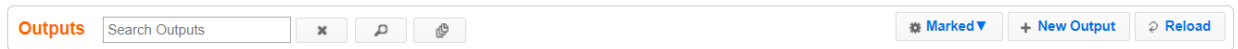
2. Click the **Marked** drop-down menu and then click **Delete**.  
The selected Output Streams are deleted.

### Start and Stop Marked Individual and Multiple Outgoing Streams

#### ➔ To Stop/Start Output Streams:

1. In the Zixi Broadcaster Outputs page, select all outputs by selecting the **Actions** checkbox, located in the header line, or select individual outputs by selecting their checkboxes.

The Marked drop-down button is displayed.



2. Click the **Marked** drop-down menu and then click **Start/Stop**.  
The selected Output Streams are started/stopped.

### Save Marked Individual and Multiple Outgoing Streams

#### ➔ To Save Output Streams:

1. In the Zixi Broadcaster Outputs page, select all outputs by selecting the **Actions** checkbox, located in the header line, or select individual outputs by selecting their checkboxes.

The Marked drop-down button is displayed.



2. Click the **Marked** drop-down menu and then click **Save**.  
The selected Output Streams are saved.

## Broadcaster Outputs

### Using Output Templates

## Reset Stats for Individual and Multiple Outgoing Streams

### ➔ To Reset Output Streams:

1. In the Zixi Broadcaster Outputs page, select all outputs by selecting the **Actions** checkbox, located in the header line, or select individual outputs by selecting their checkboxes.

The Marked drop-down button is displayed.




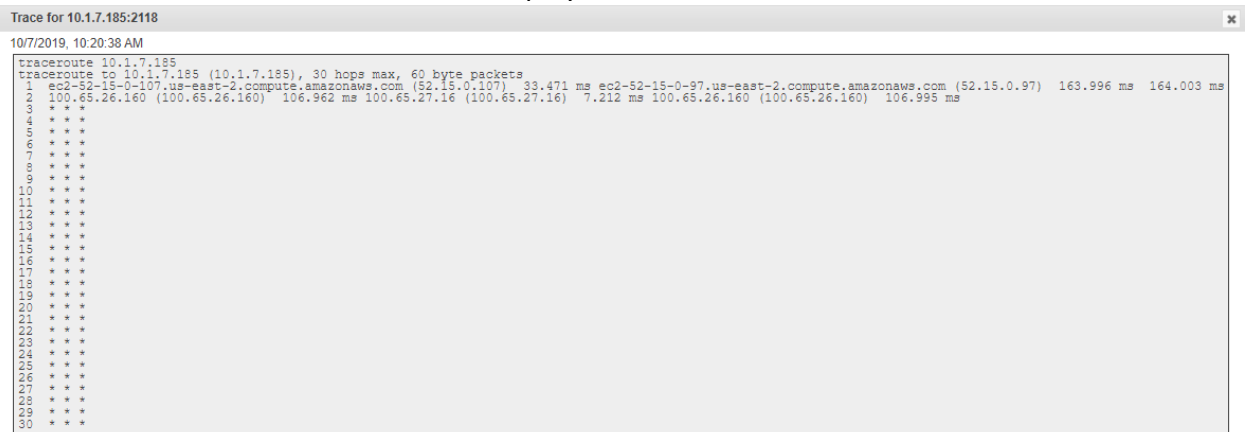
2. Click the **Marked** drop-down menu and then click **Reset**.  
The selected Output Streams are reset.

## Viewing Traceroutes

For streams sent from Zixi Broadcasters (v13 and above) to Zixi Receivers (v13 and above), you can view traceroute data for this stream.

### ➔ To view traceroute data:

1. On the **Outputs** page of Zixi Broadcaster, click on the row of the stream for which you want to view traceroutes.
2. Click the  button on the right end of the selected row and then select **Trace**.
3. The traceroute data for this stream is displayed.



[Refresh](#) [Close](#)

You can refresh the Traceroute data at any time by clicking the **Refresh** button.

## Using Output Templates

Instead of manually creating an output and then connecting the output to an input, you can define output templates with rules for automatic creation of outputs with predefined settings. The template includes an Input pattern. When an input pattern matches the pattern of an incoming input stream, the output is created automatically.

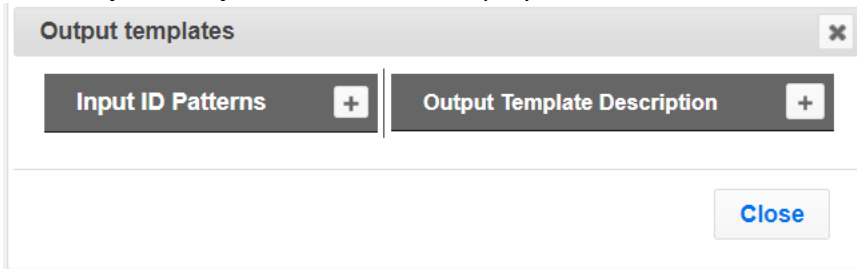
## Broadcaster Outputs

### Using Output Templates

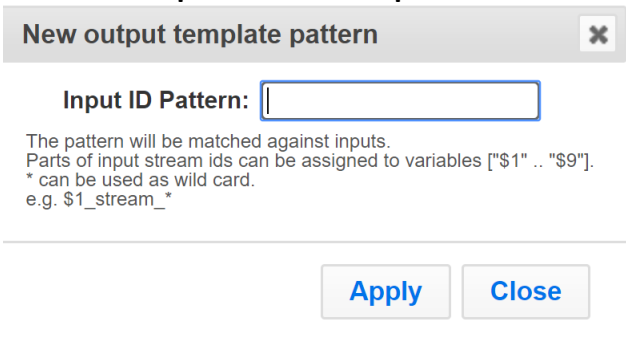
#### ➔ To create a new Output template:

1. Click the **Outputs** tab at the top of the Zixi Broadcaster administrative screen.
2. In the **Menu** bar, click the **Templates** (👤) button.

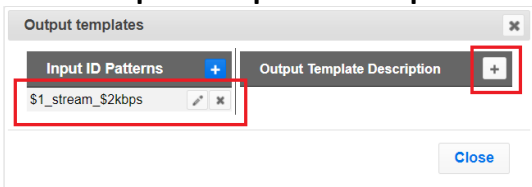
The **Output Templates** window is displayed:



3. Under **Input ID Parameters**, click the '+' button.  
The **New Output Pattern Template** window is displayed:



4. In the **Input ID Pattern** field, enter a pattern for the Stream ID of the input stream for which an output will be automatically created. This pattern will be used to identify the input stream and can include variables that are marked with the "\$" sign followed by a number (e.g. \$1, \$2, etc.) and/or with an asterisk "\*" wildcard. The patterns should also include a part of the string that requires an exact match. For example, the pattern "\$1\_stream\_\$2kbps" will match an input stream "talk\_show\_stream\_600kbps".
5. Click **Apply**.  
The newly created input ID pattern is added to the **Input ID Patterns** list.
6. From the **Input ID Patterns** list, select a pattern for which you want to relate an output template.
7. In the **Output Template Description** section click the "+" button.



## Broadcaster Outputs

### Using Output Templates

The **New Output Template** window is displayed:

**New output template**

Input ID Pattern:   
The pattern will be matched against inputs.  
Parts of input stream ids can be assigned to variables ["\$1" .. "\$9"].  
\* can be used as wild card.  
e.g. \$1\_stream\_\*

Output Name:

Show in Matrix:

UDP  RTMP  Pull  Push  TCP  HTTP Push  AWS  RIST

**Output parameters**

Host:

Port:  TTL:

Smoothing [ms]:

RTP:

Bind to IP:

Local port:

ReMuxing

Enable decryption

Enable IFB/Talkback

Ok Cancel

8. In the **Output Name** field, enter a name for the output that will be created. The name can include variables used in the **Input Pattern** field. For example, the pattern "*\$1\_ouput\_stream\_\$2kps*" will eventually create an output "*talk\_show\_output\_stream\_600kbps*", as it uses the variables that were defined in the **Input Pattern** field.
9. Select the desired output type and fill-in the relevant Output parameters according to the instructions of the relevant section:
  - **UDP Output** – Follow the instructions in the **Adding UDP Outputs** section.
  - **RTMP Output** – Follow the instruction in the **Adding RTMP and RTMPS Outputs** section.
  - **Pull Output** - Follow the instruction in the **Adding Pull Outputs** section.
  - **Push Output** - Follow the instruction in the **Adding Push Outputs** section.
  - **TCP Output** - Follow the instruction in the **Adding TCP Outputs** section.
  - **HTTP Push** - Follow the instructions in the **Adding HTTP Push Outputs** section.  
Note that the **Target URL** field can also use the same variables used in the **Input Pattern** field. For example, "[https://s3.amazon.com/zixi/\\$1\\_folder\\_\\$2kbps](https://s3.amazon.com/zixi/$1_folder_$2kbps)".



## Broadcaster Outputs

---

### Using Output Templates

- **AWS Output** - Follow the instructions in the **Adding AWS Outputs** section. Note that the **URL** field can also use the same variables used in the **Input Pattern** field. For example, "*https://s3.amazon.com/zixi/\$1\_folder\_\$2kbps*".
- **RIST Output** – Follow the instruction in the **Adding RIST Outputs** section.

10. Click **OK**.

The template is added to the **Output Template Description** list. When an input pattern will match the pattern of an incoming input stream, the defined output will be created automatically.

## Broadcaster Network Bonding

Zixi's Network Bonding feature enables you to divide a stream into several network channels by utilizing multiple NICs and links and then subsequently reuniting them at the Zixi Broadcaster side. For example, a 5 Mbps stream can be streamed using two links, one 2 Mbps link that uses a mobile network and another 3 Mbps Wifi network. The stream is automatically bonded at the Zixi Broadcaster side without any special configuration (as long as the Push input is defined and the two NICs are setup in the machine running the Zixi Broadcaster). Network Bonding can be used for load balancing between multiple links in the following configurations:

- **Zixi Feeder with Multiple NICs to a single Zixi Broadcaster NIC** - the stream is divided at the Zixi Feeder end and then bonded at the Zixi Broadcaster end. This configuration requires configuring Network Bonding in the Zixi Feeder.
- **Zixi Feeder with Multiple NICs to multiple Zixi Broadcaster NICs** - the stream is divided at the Zixi Feeder end, sent to multiple IP addresses (each Zixi Broadcaster NIC has a separate IP address), and then bonded at the Zixi Broadcaster end. This configuration requires configuring Network Bonding in the Zixi Feeder. Each destination (i.e. each Broadcaster NIC is configured in a separate network bonding link by using a different Host IP address).
- **Zixi Feeder with single NIC to Multiple Zixi Broadcaster NICs** - the stream is sent to multiple IP addresses (each Zixi Broadcaster NIC has a separate IP address), and then bonded at the Zixi Broadcaster end. This configuration requires configuring Network Bonding in the Zixi Feeder. Each destination (i.e. each Broadcaster NIC is configured in a separate network bonding link by using a different Host IP address).
- **Zixi Broadcaster with multiple NICs to another Zixi Broadcaster with a single/multiple NICs** - a Zixi Broadcaster can transfer a stream to another Zixi Broadcaster through multiple NICs at the source Broadcaster and single/multiple NICs at the destination Broadcaster. This configuration requires configuring Network Bonding in the Zixi Broadcaster (see Broadcaster Output with Network Bonding).

## Using Network Bonding for Hitless Failover

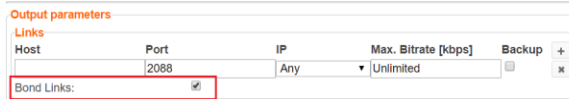
The Network Bonding feature can be used to ensure high availability of the stream. The feature ensures hit-less failover between multiple networks and network cards (i.e. network and hardware protection). As part of the configuration of the network bonding feature it is possible to set a NIC (which may be connected to a separate network) as a "Backup". In this case, the backup link will only be utilized if the stream has used all other available capacity. For example, a stream can be configured to be transferred on a WiFi network as the main link and a mobile network as a backup. If the WiFi link is compromised (either completely unavailable or limited bandwidth), the stream will be transferred, in full or partially through the backup mobile link and then bonded at the Zixi Broadcaster end, enabling uninterrupted streaming and high availability.

## Broadcaster Output with Network Bonding

A Zixi Broadcaster can push a stream to another Zixi Broadcaster that has been divided into multiple network channels by utilizing multiple NICs. The Zixi Broadcaster at the destination will subsequently reuniting them into a single input stream.

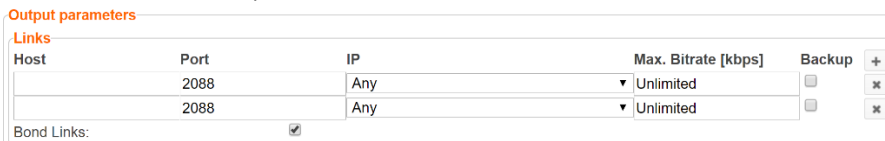
### ➔ To add network bonding to a Push Output:

1. Create a **Push Output**, by following the instructions in the [Adding Push Outputs](#) section.
2. In the **Output Parameters** section, select the **Bond Links** checkbox.



The screenshot shows the 'Output parameters' dialog box. Under the 'Links' section, there is a table with columns: Host, Port, IP, Max. Bitrate [kbps], and Backup. The first row contains: Host (empty), Port (2088), IP (Any), Max. Bitrate (Unlimited), and Backup (unchecked). Below the table, the 'Bond Links' checkbox is checked and highlighted with a red box.

3. In the **Links** section, click the **+** button to add a second link:




The screenshot shows the 'Output parameters' dialog box with two links added. The table now has two rows: the first row is identical to the previous screenshot, and the second row has: Host (empty), Port (2088), IP (Any), Max. Bitrate (Unlimited), and Backup (unchecked). The 'Bond Links' checkbox remains checked.

4. In the **Host** field, type the IP address of the destination Zixi Broadcaster.
5. In the **Port** field, type the destination port number of the Zixi Broadcaster to which you want to stream. Default: 2088
6. In the **NIC** field, select the NIC in the Zixi Broadcaster (source) that you want to use to stream.
7. In the **Bitrate Limit** field, do one of the following:
  - If you want to limit the bitrate that will be used in this link, enter the limit value in Kbps.
  - If you DO NOT want to limit the bitrate, leave the default value - "0".
8. In the **Backup** field, do one of the following:
  - If you want to use this link only if there is not enough bandwidth in the other links, select the Backup checkbox.
  - If you want to use this link in parallel to other links, DO NOT select the Backup checkbox.
9. Configure additional parameters as required.
10. Click **OK** to save the Zixi Broadcaster Output.

# Adaptive Groups

Zixi Broadcaster features the ability to create multiple bitrate profiles for a single input stream and set it as an adaptive group that can be used by a supporting client. The client will automatically switch between the various bitrate profiles of the stream according to the network conditions during playback.

After creating a group with different bitrates you can test them all out by playing them with the different 3rd party players (via the  Actions menu).

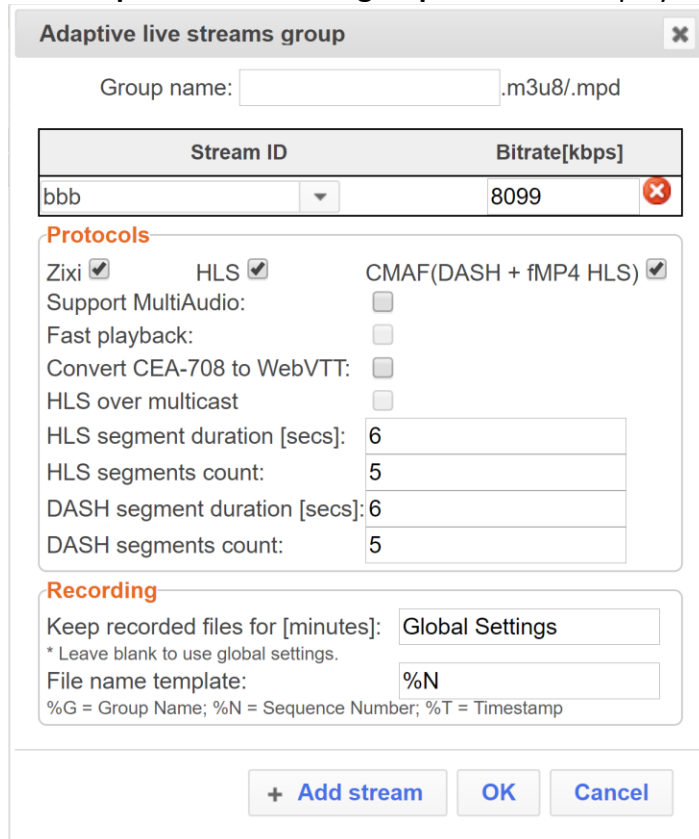
You can optionally create adaptive group templates that will automatically group streams in adaptive groups instead of defining groups in advance.

## Creating New Adaptive Groups

### ➔ To Create a New Adaptive Group:

1. In the Zixi Broadcaster administrative screen, click the **Adaptive** tab.
2. Click **Add Group**.

The **Adaptive live streams group** window is displayed:



Adaptive live streams group

Group name:

Stream ID	Bitrate[kbps]
bbb	8099

**Protocols**

Zixi  HLS  CMAF(DASH + fMP4 HLS)

Support MultiAudio:

Fast playback:

Convert CEA-708 to WebVTT:

HLS over multicast

HLS segment duration [secs]:

HLS segments count:

DASH segment duration [secs]:

DASH segments count:

**Recording**

Keep recorded files for [minutes]:

\* Leave blank to use global settings.

File name template:

%G = Group Name; %N = Sequence Number; %T = Timestamp

3. In the **Group Name** field, enter the unique string as the adaptive group's name.
4. Under **StreamID**, select an input stream from the drop-down list to create an adaptive bitrate.

## Adaptive Groups

---

### Creating New Adaptive Groups

5. Enter the desired **Bitrate[kbps]** for that stream in the adjacent field to **Stream ID**.
6. Click  to add as many streams as you want to include in the group and repeat steps 4 and 5 for every stream that is added to the group.
7. Select one or more **Protocols** that will be supported for this adaptive group. The Zixi protocol option enables devices using the Zixi SDK to view the stream using the Zixi protocol.
8. Select **Support MultiAudio** if the input streams include multiple audio channels to enable multiple separate audio channels in the adaptive group.
9. Specify the HLS/Dash segment/chunk duration and count (See details in the table below).
10. Specify the duration to **Keep recorded files for [minutes]**. If the file is left as blank, the system will use the global settings, specified in the **Settings > Live Protocols** screen in the **HLS DVR Maximum Recording Duration** field.
11. In the **File name template** field, specify the file name pattern that Zixi will use to generate when saving the recorded content.
12. Click **OK**.

The new Adaptive Group is created and appears on the Adaptive screen.

---

Parameter	Description
<b>Group Name</b>	Specify the unique identifier for this group.
<b>Stream ID</b>	Select an input stream from the drop-down list.
<b>Bitrate [kbps]</b>	Specify the bitrate for the selected stream.
<b>Protocols</b>	Select one or more <b>Protocols</b> that will be supported for this adaptive group - Zixi HLS (enables devices using the Zixi SDK to view the stream using the Zixi protocol), HLS, DASH/fMP4 HLS
<b>Support MultiAudio</b>	Select this checkbox to enable multiple separate audio channels in the adaptive group.
<b>Fast playback</b>	Select this setting to enable low latency HLS delivery. The <b>Enable HLS Fast playback</b> on the <b>Settings &gt; Live Protocols</b> screen must be enabled. Only the TS segments version of the HLS stream will be available for adaptive streaming, not the fragmented MP4 (fMP4) version. When enabled, the following playback URLs will be available: <ul style="list-style-type: none"><li>• <a href="http://&lt;server url&gt;:7777/playback.m3u8?stream=&lt;adaptive group name&gt;">http://&lt;server url&gt;:7777/playback.m3u8?stream=&lt;adaptive group name&gt;</a></li><li>• <a href="http://&lt;server url&gt;:7777/&lt;adaptive group name&gt;.m3u8">http://&lt;server url&gt;:7777/&lt;adaptive group name&gt;.m3u8</a></li></ul>
<b>Convert CEA-708 to WebVTT</b>	Extracts closed caption from video elementary stream and converts it to WebVTT format.

---

## Adaptive Groups

---

### Creating New Adaptive Groups

---

---

---

	This feature is currently available for HLS only and there is no support to styling/placement, just text.
<b>HLS Over Multicast</b>	Select this checkbox to enable multicasting of the adaptive stream (requires multicast to be enabled in the <b>Settings</b> )
<b>HLS Segment Duration [sec]</b>	Specify the duration in seconds of each segment/chunk in the HLS playlist. You can specify sub-second durations (e.g. 0.5). (Default: 6 seconds)
<b>HLS Segments Count</b>	Specify the number of segments in the HLS playlist.
<b>DASH Segment Duration [sec]</b>	Specify the duration in seconds of each segment/chunk in the DASH adaptive group. You can specify sub-second durations (e.g. 0.5). (Default: 6 seconds)
<b>DASH Segments Count</b>	Specify the number of segments in the DASH adaptive group.
<b>Recording</b>	
<b>Keep recorded files for [minutes]</b>	Specify the maximum time for storing the recording. If the file is left as blank, the system will use the global settings specified in the <b>Settings &gt; Live Protocols</b> screen in the <b>HLS DVR Maximum Recording Duration</b> field.
<b>File name template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>%S=stream id - optional.</li><li>%Y=year - optional</li><li>%M=month - optional</li><li>%D=day - optional</li><li>%T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>




---

---

## Adding Streams to Existing Adaptive Groups

You can add additional streams to existing adaptive groups.


### ➔ To Add Streams to an Adaptive Group:

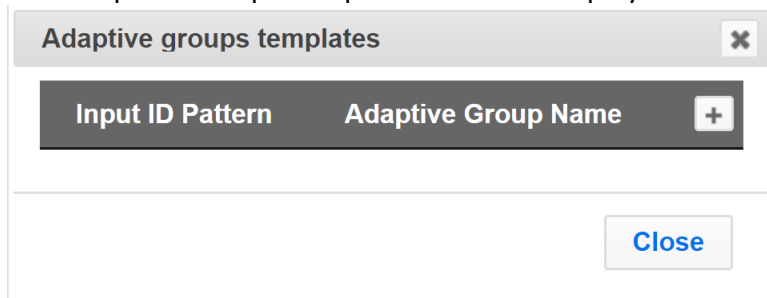
1. In the Zixi Broadcaster administrative screen, click the **Adaptive** tab.
2. Select the Group to which you want to add a stream, and click  button, and then **Edit**. The **Adaptive live streams group** window is displayed.
3. Click  button for as many streams as you want to include in the group. Another Stream row will be added in the above table to which a desired **Bitrate** must be specified. When clicking on the red  at the end of any of the stream rows, the stream will be deleted from the group.
4. Click **OK**. The group has been modified accordingly.

## Adaptive Groups Templates

Instead of defining adaptive groups in advance and then relating specific input streams to these groups, you can define templates with rules for automatic grouping of streams in adaptive groups. Every time a new input stream, which matches the conditions of the template, is added, it will be automatically added to the adaptive group that was defined in the template. Specific streams are identified by naming patterns and added to the group by setting static naming patterns and bitrates or dynamic patterns.

➔ **To create a new adaptive group template:**

2. Click the **Adaptive** tab at the top of the Zixi Broadcaster administrative screen.
3. In the Menu bar, click the **Templates** (  ) button.  
The Adaptive Groups Templates screen is displayed:





## Adaptive Groups

### Adaptive Groups Templates

- Click the + button.

The **Adaptive Group Template** window is displayed:

**Adaptive group template** [X]

Input ID Pattern will be matched against inputs.  
Parts of input stream ids can be assigned to variables ["\$1" .. "\$9"].  
\* can be used as wild card.  
e.g. \$1\_stream\_profile\_\*

Input ID Pattern:

---

Adaptive Group Name:

Variables from Input ID pattern can be used to form adaptive group name.  
e.g. \$1\_adaptive\_stream

**Adaptive group properties**

Zixi       HLS       DASH

HLS over multicast

Record

Keep recorded files for [minutes]:

\* Leave blank to use global settings.

File name template:

%G = Group Name; %N = Sequence Number; %T = Timestamp

---

**Adaptive group stream patterns**

Streams expected in adaptive group may connect later.

Dynamic streams acquisition       Static streams set

Stream pattern      Bitrate [kbps]

Apply      Close

- In the **Input ID Pattern** field, enter a pattern for the Stream ID of the input stream that will be automatically added to newly created adaptive group. This pattern will be used to identify the input stream and can include variables that are marked with the "\$" sign followed by a number (e.g. \$1, \$2, etc.) and/or with an asterisk "\*" wildcard. The patterns should also include a part of the string that requires an exact match. For example, the pattern "\$1\_stream\_\$2kbps" will match an input stream "talk\_show\_stream\_600kbps". String parts that were matched against \$X variables, can be reused in other settings, however, string parts that were matched against an asterisk wildcard cannot be reused in other settings.
- In the **Adaptive Group Name** field, enter a string that will be used as the name of the adaptive group. You can use the \$X variables that were used in the Input ID Pattern field above. For example, based on the above example, if you write "\$1\_group", the group name will be "talk\_show\_group".
- Under the **Adaptive Group Properties**, select the checkboxes of the protocols with which these streams will be transmitted (Zixi, HLS, DASH, HLS over multicast).
- If you want to record an HLS stream, select the **Record** checkbox and in the **Keep recorded files for [minutes]**, specify the maximum time (in minutes) to record. Default:

## Adaptive Groups

### Adaptive Groups Templates

the time defined in the global settings, specified in the **Settings > Live Protocols** screen in the **HLS DVR Maximum Recording Duration** field.

- Under **Adaptive Group Stream Patterns**, select the method in which the streams in the adaptive group will be created:
  - Dynamic Streams Acquisition:** this method will automatically add input streams that match the defined **Input ID Pattern** and give them a bitrate value based on the **Bitrate Pattern**. This option is based on the assumption that the bitrate value is part of the input stream's ID. Based on the example above, by entering the value "\$2" in the **Bitrate Pattern** field, the system will automatically add the incoming "talk\_show\_stream\_600kbps" stream, while giving it a 600 Kbps Bitrate setting (according to the definition in the **Input ID Pattern** field, the \$2 was assigned a value of 600kps). If another stream that matches the input ID patterns comes in (e.g. talk\_show\_stream\_700kbps") with a different bitrate, for example, 700kbps, this stream will be also added to the same adaptive groups, however the bitrate setting will be 700 kbps.

The screenshot shows a configuration window titled "Adaptive group stream patterns". It contains the following elements:

- A warning message: "Streams expected in adaptive group may connect later."
- Two radio buttons: "Dynamic streams acquisition" (selected) and "Static streams set".
- A note: "Any stream matching 'Input ID Pattern' will be added."
- A text input field labeled "Bitrate pattern:" containing the value "\$2".

#### To add a Dynamic Streams Acquisition bitrate pattern:

- In the **Bitrate Pattern** field add the pattern as described above.
- Click **Apply**.

The new template is added to the **Adaptive Groups Templates** list:

The screenshot shows a window titled "Adaptive groups templates" with a table of templates. The table has two columns: "Input ID Pattern" and "Adaptive Group Name".

Input ID Pattern	Adaptive Group Name	
\$1_stream_\$2kbps	\$1_adaptive_stream	

At the bottom right of the window is a "Close" button.

- Static Stream Set:** if the input streams do not have the bitrate value in their ID names, you can manually define the ID and the bitrate for each stream that will be included in the group. You can define streams that are not currently connected, but will be connected at a later stage.

## Adaptive Groups

### Performing Actions on Adaptive Groups

**Adaptive group stream patterns**  
Streams expected in adaptive group may connect later.

Dynamic streams acquisition  Static streams set

Stream pattern	Bitrate [kbps]	
<input type="text" value="comedy_HD"/>	<input type="text" value="512"/>	<input type="button" value="+"/>
		<input type="button" value="x"/>

#### To add a Static Steam Set:

1. Click the + button
2. In the **Stream Pattern** field, enter a pattern for the stream ID (you can use the variables used in the Input ID Pattern field and add additional variables as needed). This pattern will be used to identify the stream and add it to the group (e.g. "comedy\_HD").
3. In the **Bitrate [Kbps]** field, enter the Bitrate value of the stream.
4. Click the + button for as many streams as you want to include in the group and repeat the steps above for every stream that is added to the group.
5. Click **Apply** to save all changes.

The new template is added to the **Adaptive Groups Templates** list:

Adaptive groups templates		
Input ID Pattern	Adaptive Group Name	
\$1_stream_\$2kbps	\$1_adaptive_stream	<input type="button" value="+"/>
		<input type="button" value="edit"/> <input type="button" value="x"/>



the group will be active only when all the streams defined will be connected.

## Performing Actions on Adaptive Groups

Every adaptive group has a shortcut button that can be used to perform relevant actions.

### Stopping/Starting an Adaptive Group


You can start or stop the stream(s) in the adaptive group at any time.

## Adaptive Groups

---


### Performing Actions on Adaptive Groups

#### ➔ To Stop/Start the Adaptive Group:

1. In the Zixi Broadcaster **Adaptive Groups** page, click the adaptive group row that you want to start/stop.
2. Click the  button on the right end of the selected row and then **Start/Stop**. The status will change to Connected/Stopped accordingly.


### Editing an Existing Adaptive Group

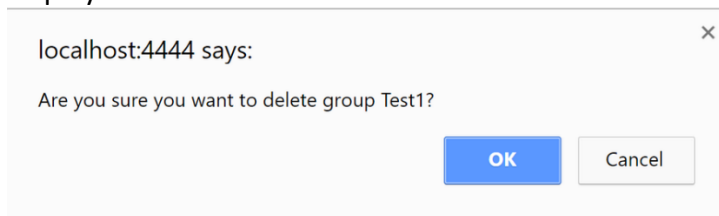
#### ➔ To Edit the Configuration of an Adaptive Group:

1. In the Zixi Broadcaster **Adaptive Groups** page, click the adaptive group row that you want to edit.
2. Click the  button on the right end of the selected row and then **Edit**. The Adaptive live stream group screen is displayed. You can modify any of the settings.
3. Click **OK**. The new settings will take effect.

### Deleting and Adaptive Group

#### ➔ To delete an Adaptive Group:

1. In the Zixi Broadcaster **Adaptive Groups** page, click the adaptive group row that you want to delete.
2. Select the Group to which you want to add a stream, and click  button, and then **Delete**. The message "Are you sure you want to delete group <group\_name>?" is displayed.



3. Click **OK**. The selected row adaptive group will be deleted.

### Play Dash


Zixi Broadcaster can generate links to play the Adaptive Group stream in MPEG DASH format.

#### ➔ To Play MPEG DASH:

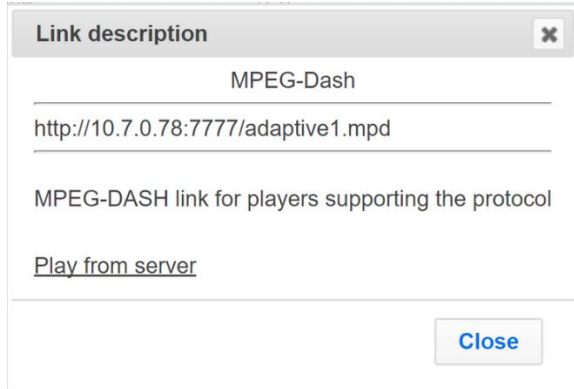
1. In the Zixi Broadcaster **Adaptive Groups** page, click the adaptive group row that you want to playback.

## Adaptive Groups

### Performing Actions on Adaptive Groups

2. Select the Group to which you want to add a stream, and click  button, and then click **Play Dash**.

The following window is displayed:




3. Do one of the following:
  - If you have a locally installed MPEG DASH compatible player - click the link to stream the data to your player.
  - If you DO NOT have a player - click **Play from server**. The stream will open in a web-based player.
4. Click **Close** when finished.

## Play fMP4

Zixi Broadcaster can generate links to play the Adaptive Group stream in fMP4 HLS format.

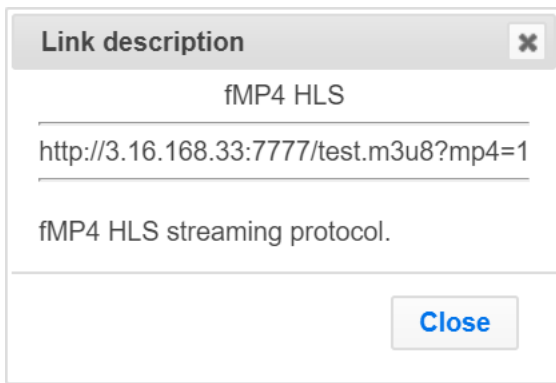
### ➔ To Play fMP4 HLS:

1. In the Zixi Broadcaster **Adaptive Groups** page, click the adaptive group row that you want to playback.
2. Select the Group to which you want to add a stream, and click  button, and then click **Play fMP4**.

The following window is displayed:

## Adaptive Groups

### Performing Actions on Adaptive Groups




3. Copy the link that is generated at the bottom of the window and paste it into a player/browser for playback.
4. Click **Close** when finished.

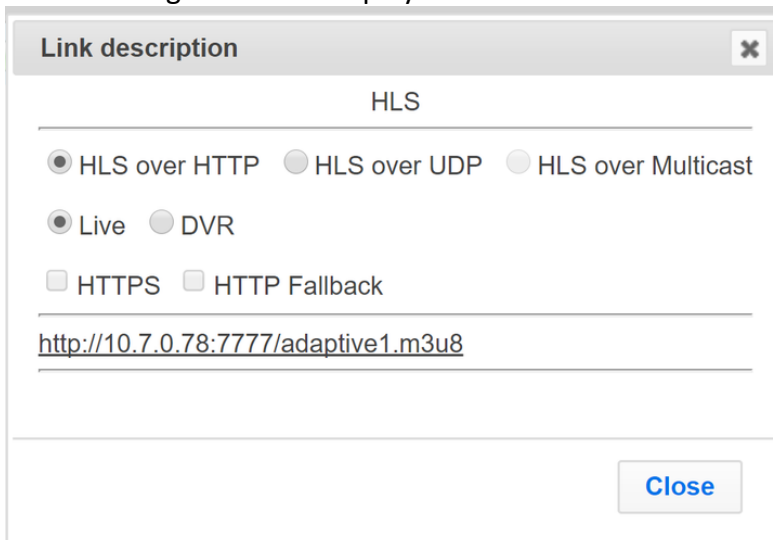
## Play HLS

Zixi Broadcaster can generate links to play the Adaptive Group stream in HLS format over HTTP, UDP, or Multicast (requires Multicast enabled).

### ➔ To Play HLS:

1. In the Zixi Broadcaster **Adaptive Groups** page, click the adaptive group row that you want to playback.
2. Select the Group to which you want to add a stream, and click  button, and then click **Play HLS**.

The following window is displayed:



3. Do one of the following:

## Adaptive Groups

### Performing Actions on Adaptive Groups

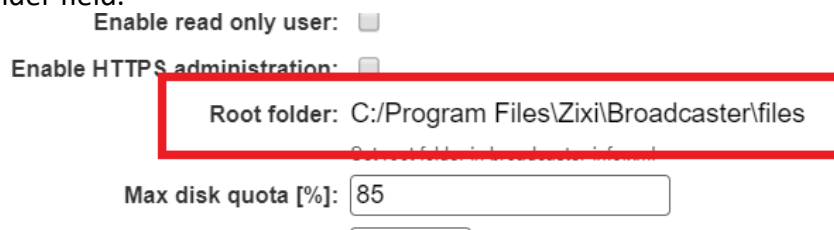
- **HLS over HTTP** - streams the adaptive stream over HTTP. Select **Live** to stream a live stream or select **DVR** to select a recorded stream. To stream from an encrypted HTTPS link, select the **HTTPS** checkbox. To stream from a secondary/backup server, select the **HTTP Fallback** checkbox.
  - **HLS over UDP** - streams the adaptive stream over UDP. Select **Live** to stream a live stream or select **DVR** to select a recorded stream. To stream from an encrypted HTTPS link, select the **HTTPS** checkbox. To stream from a secondary/backup server, select the **HTTP Fallback** checkbox.
  - **HLS over Multicast** - streams the adaptive stream over UDP. Select **Live** to stream a live stream or select **DVR** to select a recorded stream. To stream from an encrypted HTTPS link, select the **HTTPS** checkbox. To stream from a secondary/backup server, select the **HTTP Fallback** checkbox.
4. Copy the link that is generated at the bottom of the window and paste it into a player/browser for playback.
  5. Click **Close** when finished.

## Set Preroll

A preroll transport stream file is a video/audio file that is played before the adaptive group video is played. Before setting the preroll, you should make sure that the file is available for VOD and that VOD is enabled in the File Transfer and VOD settings.

### ➔ To set a Pre-roll:

1. Upload the Pre-roll TS File to the designated Root folder on the Zixi Broadcaster server. To find your Root folder location, go to Settings > General. The Root path is specified in the Root Folder field.



2. In the **Settings > File Transfer & VOD** screen, select the **VOD** checkbox. The following window is displayed:




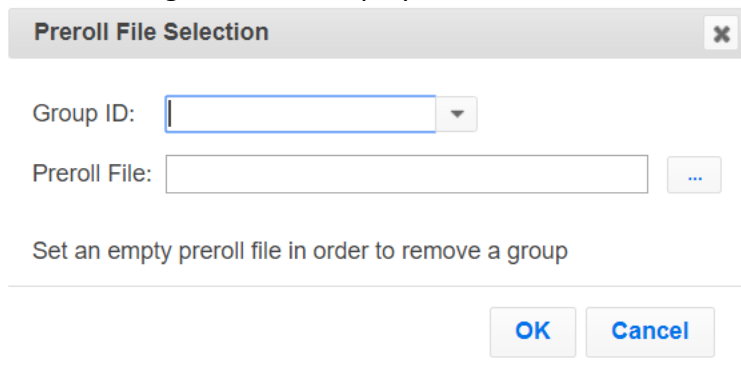
3. In the Zixi Broadcaster Adaptive Groups screen, click the desired adaptive group for which you want to add the preroll.

## Adaptive Groups

---

### Performing Actions on Adaptive Groups

- Click the  button on the selected row and then click **Set Preroll**. The following window is displayed:



The dialog box titled "Preroll File Selection" contains the following elements:

- A "Group ID" field with a dropdown arrow.
- A "Preroll File" field with a "Browse" button (three dots).
- A note: "Set an empty preroll file in order to remove a group".
- "OK" and "Cancel" buttons at the bottom.

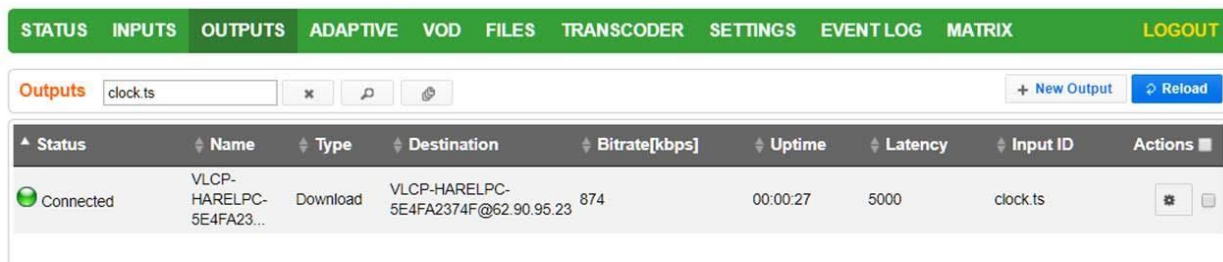
- In the **Group ID** field, select the adaptive group for which you want to add the preroll.
- In the **Preroll File** field, click the **Browse** button and select the desired file.
- Click **OK**.



# VOD (Video on Demand)

The VOD feature enables you to upload stream-able (.ts and mp4) files to the file system of the server on which Broadcaster is installed as well as make a previously recorded input stream available on demand (VOD). The VOD content is accessed through a local folder, where the Zixi Broadcaster is located. This local folder can be optionally mapped through a symbolic link or shortcut to a remote folder. The Broadcaster creates a unique URL for each file.

The streams can be played back using VLC (with Zixi Plugin for Zixi streams) and/or streamed to a Zixi Receiver or to a Zixi Broadcaster. During playback, an Output (with output type – **Download**) will be created automatically on the Source Zixi Broadcaster, as shown below.



Status	Name	Type	Destination	Bitrate[kbps]	Uptime	Latency	Input ID	Actions
Connected	VLCP-HARELPC-5E4FA23...	Download	VLCP-HARELPC-5E4FA2374F@62.90.95.23	874	00:00:27	5000	clock.ts	[Settings] [Close]

## VOD Activation

If you have a VOD-enabled license you will still need to activate the VOD feature in the **Settings** screen.

### ➔ To activate the VOD feature:

1. In the **Settings** page, click **File Transfer & VOD**.
2. Select the **VOD** checkbox.



HLS Indexing:

MPEG-DASH Indexing:

**VOD:**

Encrypt downloads:

Max download speed [kbps]:  ~100.0 Mbit/s

3. Click **Apply**.

## VOD Files Location

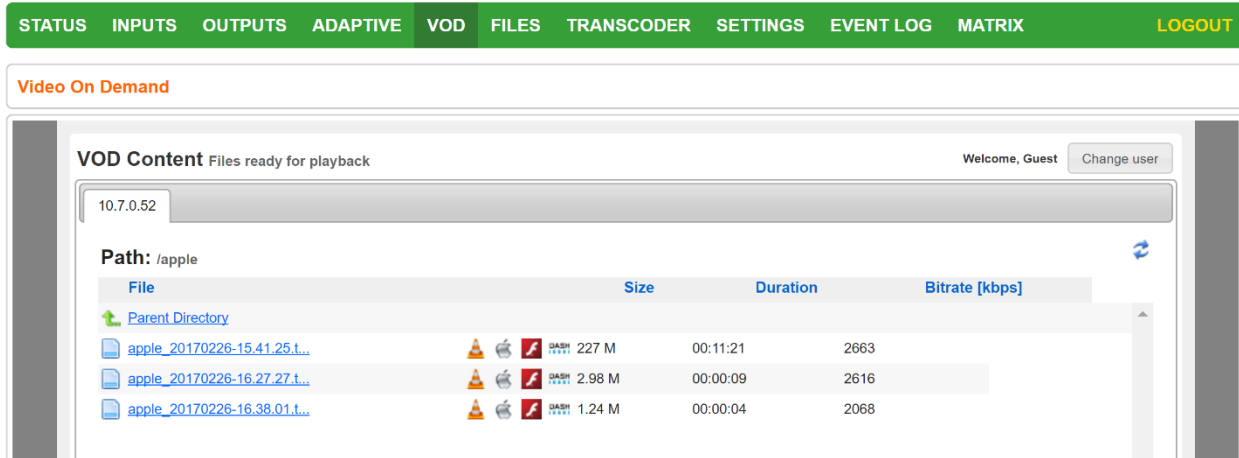
The VOD files are located in the following location by default -

<Broadcaster\_Installation\_Folder>\Zixi\Broadcaster\files

## VOD (Video on Demand)

### VOD Files Location

The contents of this folder are displayed in the **VOD** screen on the Zixi Broadcaster GUI –



The VOD screen displays the following information:

Parameter	Description
-----------	-------------

<b>File</b>	Name of the file or folder. If in file level (not folder level), click the file name to preview the stream in the Matrix screen.
-------------	--

The following icons are displayed:



- **VLC Playback** – click this icon to see the VLC playback URL and additional details, such as VLC plugin download link, and link to stream the file.



- **HLS Playback** – click this icon to see the following three HLS playback URLs: local HLS playback (not Zixi protected), HLS playback using Zixi Proxy (Zixi Protected), HLS playback using Zixi Proxy through HTTPS (Zixi protected). The Zixi Proxy links require the installation of Zixi Proxy software in the receiving device/machine. The dialog includes links for downloading the Zixi Proxy for Windows and Mac.

<Icons>



- **HDS Playback** – click this icon to see the Adobe HTTP Dynamic Streaming Protocol playback URL. The Play from server link, plays back the stream directly from the server.



- **Flash Playback** – click this icon to see the following three Flash playback URLs: local Flash playback (not Zixi protected), Flash playback using Zixi Proxy (Zixi Protected), Flash playback using Zixi Proxy through HTTPS (Zixi protected). The Zixi Proxy links require the installation of Zixi Proxy software in the receiving device/machine. The dialog includes links for downloading the Zixi Proxy for Windows and Mac.



- **Dash Playback** - click this icon to see the MPEG-DASH playback URL. The Play from server link, plays back the stream directly from the server.

<b>Size</b>	The size of the file.
-------------	-----------------------

<b>Duration</b>	The duration of the playback.
-----------------	-------------------------------

## VOD (Video on Demand)

---

### Adding Files to VOD

---

---

<b>Bitrate [kbps]</b>	The bitrate of the file in Kbps.
---------------------------	----------------------------------

---

---

## Changing the Files folder location

### ➔ To change the Files folder location:

1. In the Zixi Broadcaster installation folder  
(`<Broadcaster_Installation_Folder>\Zixi\Broadcaster\`) open the following file: **broadcaster-info.xml**.
2. Edit the `<Folder>` parameter with the desired path.

```
<?xml version="1.0" encoding="utf-8"?>
<ROOT_SETTINGS>
  <FOLDER>C:\Program Files\Zixi\Broadcaster\files</FOLDER>
  <MANAGED>>false</MANAGED>
  <MANAGEMENT-ENABLED>>false</MANAGEMENT-ENABLED>
</ROOT_SETTINGS>
<TUNNEL>
  <ALLOW>>false</ALLOW>
  <MANAGED>>false</MANAGED>
  <MANAGEMENT-ENABLED>>false</MANAGEMENT-ENABLED>
</TUNNEL>
```

3. Save the file.

## Adding Files to VOD

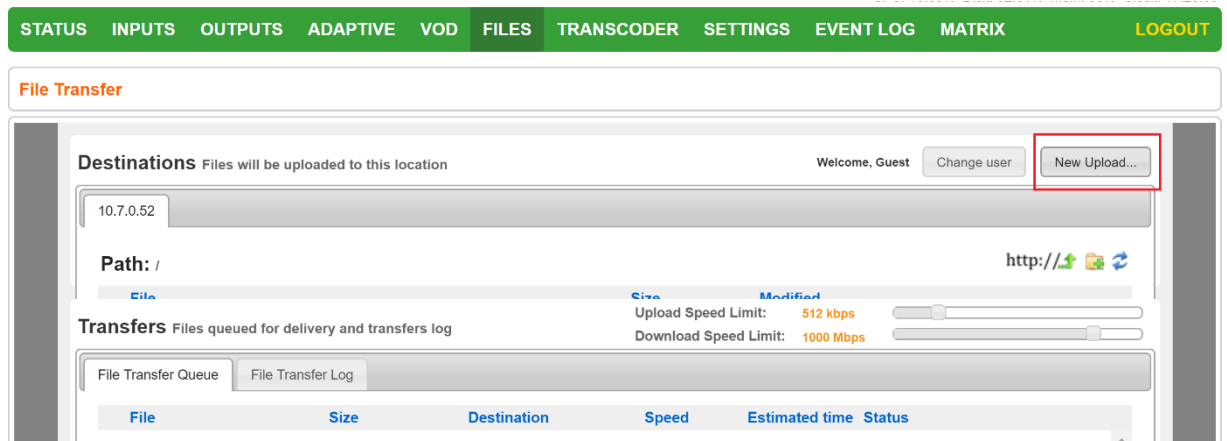
There are two ways to add files into the VOD Files folder:

- **Adding files to the folder** – using the operating system on which Zixi Broadcaster is installed, add the desired files to the Files folder  
(`<Broadcaster_Installation_Folder>\Zixi\Broadcaster\Files`)

## VOD (Video on Demand)






### Playing back VOD content

- **Uploading using the Broadcaster GUI** – on the **Files** screen, click the **New Upload** button, browse to the desired file location, select the file and click **Upload**.



## Playing back VOD content

The VOD content can be played back using VLC (with Zixi Plugin for Zixi streams) and/or streamed to a Zixi Receiver or to a Zixi Broadcaster by using the following methods:

- **URL (Static)** – entering a specific URL of a file. To find the URL of a specific file, go to the VOD screen, navigate to the specific file, and click on the relevant playback icon:
  -  - **VLC Playback** – click this icon to see the VLC playback URL and additional details, such as VLC plugin download link, and link to stream the file.
  -  - **HLS Playback** – click this icon to see the following three HLS playback URLs: local HLS playback (not Zixi protected), HLS playback using Zixi Proxy (Zixi Protected), HLS playback using Zixi Proxy through HTTPS (Zixi protected). The Zixi Proxy links require the installation of Zixi Proxy software in the receiving device/machine. The dialog includes links for downloading the Zixi Proxy for Windows and Mac.
  -  - **HDS Playback** – click this icon to see the Adobe HTTP Dynamic Streaming Protocol playback URL. The Play from server link, plays back the stream directly from the server.
  -  - **Flash Playback** – click this icon to see the following three Flash playback URLs: local Flash playback (not Zixi protected), Flash playback using Zixi Proxy (Zixi Protected), Flash playback using Zixi Proxy through HTTPS (Zixi protected). The Zixi Proxy links require the installation of Zixi Proxy software in the receiving device/machine. The dialog includes links for downloading the Zixi Proxy for Windows and Mac.
  -  - **Dash Playback** - click this icon to see the MPEG-DASH playback URL. The Play from server link, plays back the stream directly from the server.

## VOD (Video on Demand)

---

### Indexing VOD content

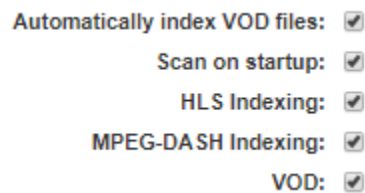
- **API (Dynamic)** – you can use the Zixi API to implement dynamic access to the VOD content. For example, you can create an application that enables navigation throughout the VOD library (including folders and sub-folders) to access files in a dynamic fashion. The API GET request is: <http://<broadcaster location><web server port>/vod ls.json>

## Indexing VOD content

Zixi Broadcaster can create m3u8 (HLS) and mpd (mpeg-dash) manifests for VOD files. When the **Automatically index VOD files** setting is enabled, Zixi Broadcaster will index the VOD files in the various VOD folders. In this case, Zixi Broadcaster will automatically monitor any file and directory structure changes (under "files" folder only). Any file that is added will be automatically indexed. In the same manner, any file that is deleted, the correspondent files in the index (.m3u8 and/or /mpd) will be removed as well. In some cases, the indexing of VOD files can affect performance. By enabling or disabling the **Automatically index VOD files** setting, you can decide when to index the VOD files in order to ensure optimal performance during peak times.

### ➔ To activate the VOD indexing:

1. In the **Settings** page, click **File Transfer & VOD**.
2. Select the **Automatically index VOD files** checkbox.



3. Select **Scan on startup** to enable indexing of the files in the VOD folder that have not been indexed during startup of the Zixi Broadcaster.
4. Select **HLS indexing** to enable indexing of HLS streams. The indexing creates an M3U8 playlist file.
5. Select **MPEG-DASH indexing** to enable indexing of MPEG-DASH streams. The indexing creates a .mpd (MPEG-DASH Media Presentation Description) file.
6. Click **Apply**.

# Accelerated File Transfer

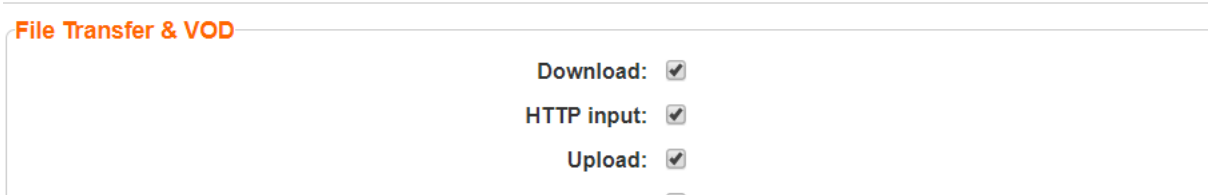
Zixi Broadcaster features the ability to transfer files between two Zixi components (Zixi Broadcaster to Zixi Receiver or to another Zixi Broadcaster), using the Zixi protocol. The transfer can also be to a non-Zixi component with a Zixi Proxy or with a Zixi Plugin. This enables the reliable and fast transfer of large files over a wireless/wifi connections.

## File Transfer Configuration

Before using the file transfer feature, you will need to activate the file transfer feature and configure the following file transfer settings.

➔ **To activate and configure the file transfer feature:**

1. In the **Settings** page, click **File Transfer & VOD**.



2. Select the **Download** checkbox to enable the download of files from the current Zixi Broadcaster.
3. Select **HTTP Input** checkbox to enable HTTP forwarding.
4. Select the **Upload** checkbox to enable the uploading of files by registered users to the current Zixi Broadcaster.
5. Specify additional optional **file transfer setting parameters** (see File Transfer Setting table below).
6. Click **Apply**.

## File Transfer Settings

Parameter	Description
<b>Download</b>	Enables the file transfer download from this Zixi Broadcaster.
<b>HTTP Input</b>	Enables HTTP acceleration mode. In this mode, instead of specifying a file location on the server's hard disk, the Zixi Broadcaster receives an HTTP address from which it will retrieve the file and then transfer it to the Client.
<b>Upload</b>	Enables the file transfer upload to this Zixi Broadcaster server.
<b>Encrypt Downloads</b>	Enables the Zixi auto encryption for all file transfer downloads.

## Accelerated File Transfer

---

### Uploading Files for File Transfer

---

---

<b>Max download speed [kbps]</b>	Specifies the maximal download speed in Kbps for file transfer.
<b>Max upload speed [kbps]</b>	Specifies the maximal upload speed in Kbps for file transfer.
<b>Initial bitrate [kbps]</b>	Specifies the initial download bitrate. The recommended bitrate can be the bitrate of the stream.
<b>HTTP/Origin cache size [MB]</b>	This setting is relevant for HTTP acceleration mode and/or Origin/Edge architecture, where the Zixi Broadcaster retrieves the file from another Origin or HTTP server and then delivers it to the Client. This setting determines the cache size allocation on the Zixi Broadcaster. The cache is used to accelerate the delivery of files that have been previously requested.

---

---

## Uploading Files for File Transfer

You can upload files to the server on which Zixi Broadcaster is installed to make them available for download using Zixi protocol. The location of the files that will be uploaded is relative to the Root folder that was specified in the **Settings > General** screen.



Before you upload a file, make sure Zixi Proxy is installed on your computer. If it is not installed, you will be prompted with a notification and a link for download and installation.

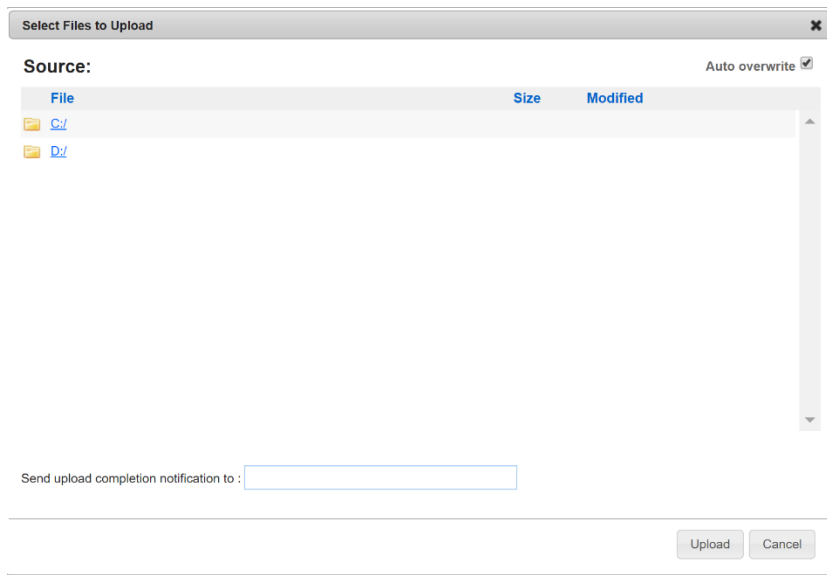
### ➔ To upload a file for file transfer:

1. In Zixi Broadcaster, click the **Files** tab.
2. Click **New Upload**.

The **Select Files to Upload** dialog opens:


## Accelerated File Transfer

### Downloading Files

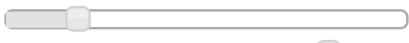
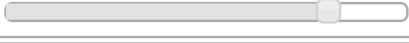


3. Navigate to the path in which the files is located and select the file.
4. For very large files, it is possible to receive an email notification when the upload process was completed. To enable this feature, type the desired email address, or multiple email addresses separated by commas in the **Send upload completion notification to** field.
5. Click **Upload**.

The file will be uploaded using the Zixi protocol. The upload process is displayed at the bottom of the screen (**Files Transfer Queue** tab). When the upload is complete, it will be recorded in the **File Transfer Log** tab.



You can limit the upload speed by dragging the **Upload Speed Limit** slider.

Upload Speed Limit:	512 kbps	
Download Speed Limit:	1000 Mbps	

The overall maximal limit determined by the **Max upload speed [kbps]** setting in the **Settings > File Transfer & VOD** screen.

## Downloading Files

The files can be downloaded using the Zixi Protocol, through the Zixi Broadcaster UI.

### ➔ To download a file:

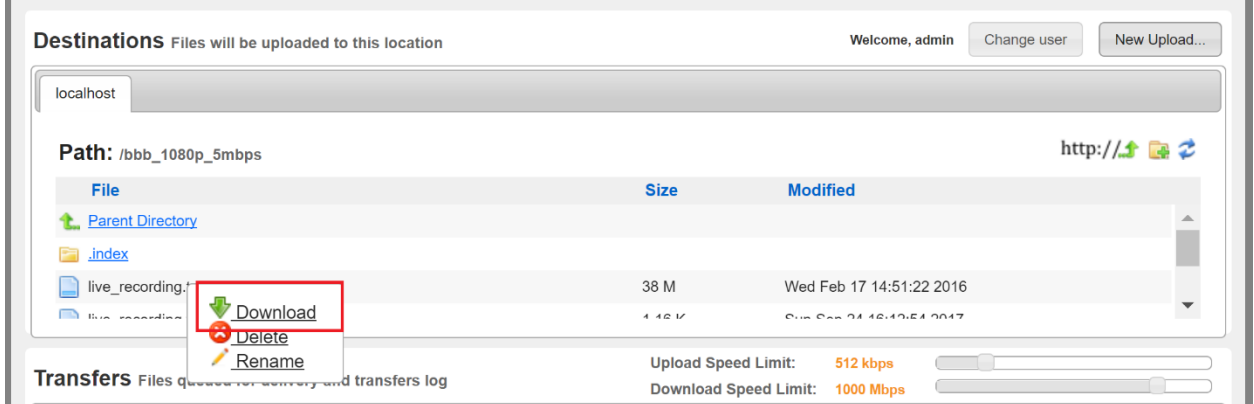
1. In Zixi Broadcaster, click the **Files** tab.
2. Under **Destinations**, navigate to the desired file.



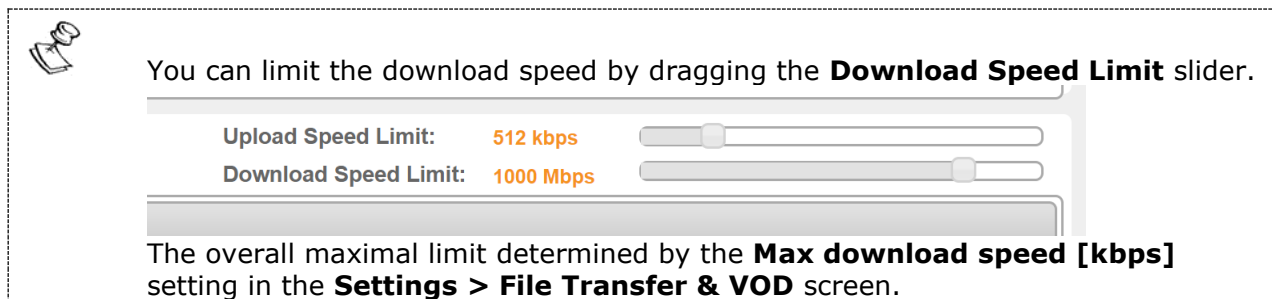
## Accelerated File Transfer

### Generating Accelerated URLs

3. Hover your mouse over the desired file and click **Download**.



4. In the **Select Download Destination** window, select the local folder as a destination for the file download.  
The file will be downloaded using the Zixi protocol. The download process is displayed at the bottom of the screen (**Files Transfer Queue** tab). When the download is complete, it will be recorded in the **File Transfer Log** tab.



## Generating Accelerated URLs


Instead of downloading files from the server on which Zixi Broadcaster is installed, it is possible to use Zixi Broadcaster to retrieve the file from a remote HTTP server. Zixi Broadcaster converts the remote server URL to an Accelerated URL. By using this accelerated URL to download the file, Zixi Broadcaster will retrieve the file from the remote server and use the Zixi Protocol to transfer the file to the Client.

### ➔ To generate an accelerated URL:

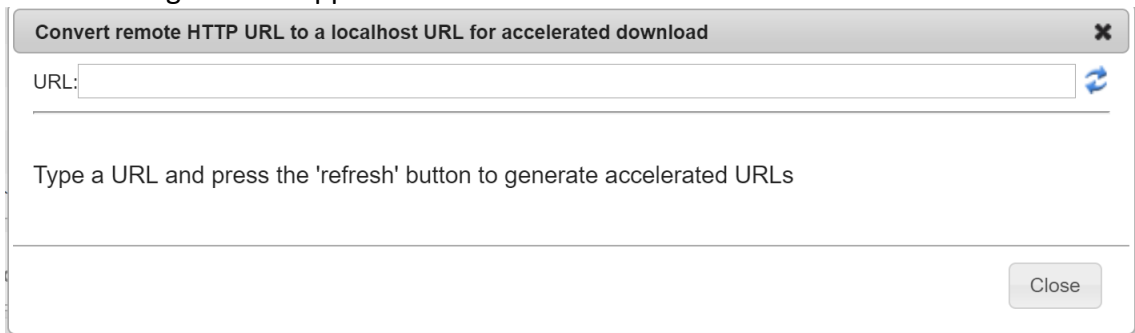
1. In Zixi Broadcaster, click the **Files** tab.


## Accelerated File Transfer

### Creating a New Folder

2. Under **Destinations**, click the  icon.

The following window appears:



3. In the **URL** field, type the URL of file in the remote HTTP server.
4. Click the  icon.

Two URLs are generated:




The top URL is an HTTP URL and the bottom is an HTTPS URL.

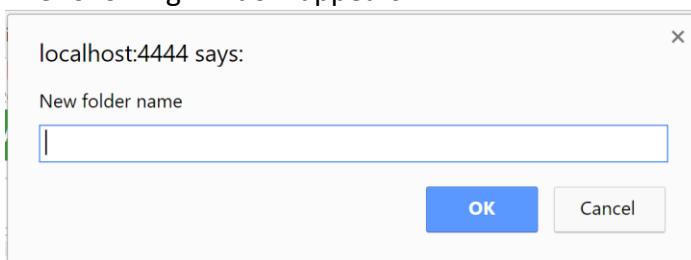
## Creating a New Folder

You can use the Zixi Broadcaster UI to create folders on the server on which Zixi Broadcaster is installed.

### ➔ To create a new folder:

1. In Zixi Broadcaster, click the **Files** tab.
2. Under **Destinations**, click the  icon.

The following window appears:



## Accelerated File Transfer

### Renaming a File/Folder

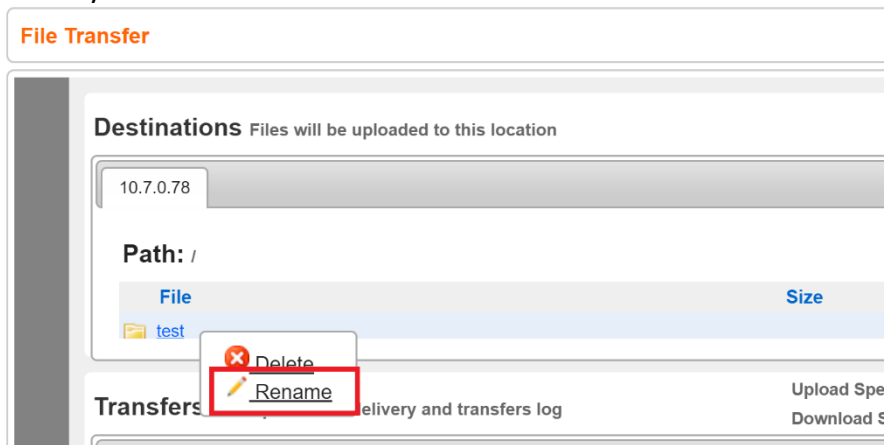
3. In the **New folder name** field, type a name for the folder.  
The folder is created.

## Renaming a File/Folder

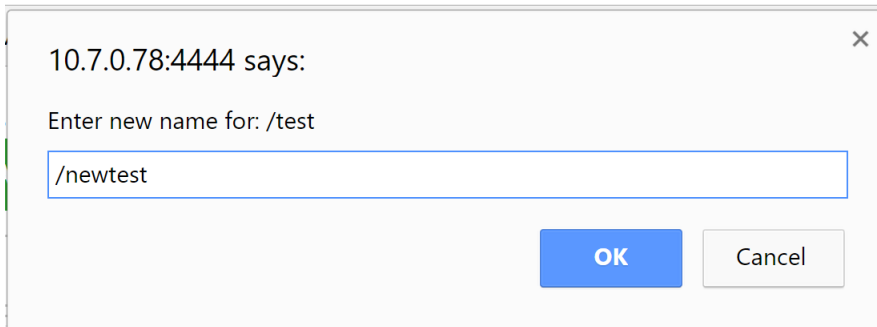
You can rename a folder or a file through the Zixi Broadcaster UI.

### ➔ To rename a folder/file:

1. In Zixi Broadcaster, click the **Files** tab.  
Under **Destinations**, navigate to the desired file.
2. Hover your mouse over the desired file and click **Rename**.



3. In the field, enter the new folder/file name. The location of the files/folder is relative to the Root folder that was specified in the **Settings > General** screen.



4. Click **OK**.

## Deleting an Existing File/Folder

You can delete a folder or a file through the Zixi Broadcaster UI.

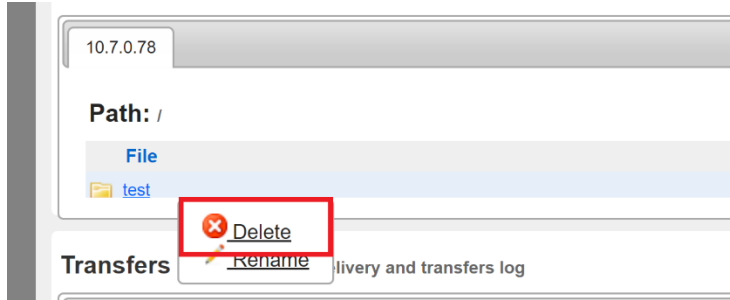
## Accelerated File Transfer

---

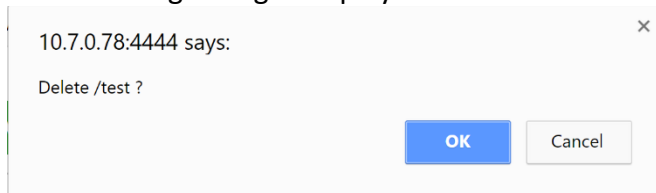
### Deleting an Existing File/Folder

#### ➔ To delete a folder/file:

1. In Zixi Broadcaster, click the **Files** tab.
- Under **Destinations**, navigate to the desired file.
5. Hover your mouse over the desired file and click **Delete**.



The following dialog is displayed:



6. Click **OK**.

# Transcoder

Zixi Broadcaster offers live transcoding, enabling users to create multiple bitrate streams from a single input stream. The transcoding converts the video and audio live streams to a variety of profiles and bitrates. SCTE-35 markers will be preserved by the transcoder.

The transcoder is typically employed in the following cases:

- Where a target device does not support the source format.
- Where a target has limited storage capacity that requires a reduced file size.
- To convert incompatible obsolete data to a better-supported or modern format.
- You can add video (H.264, H.265, MPEG2, Raw) and audio (AAC, OPUS, Raw) profiles.

### Transcoding Templates

Instead of manually selecting input streams to be transcoded, you can optionally define templates with rules for automatic transcoding of input streams according to the settings of a pre-defined Transcoding Template.

### NDI Transcoding

For details about NDI transcoding, see [Configuring NDI Streams](#) section.

### WebRTC Transcoding

For details about WebRTC transcoding, see section [Configuring WebRTC Streams](#).

## Creating New Profiles

In order for the Broadcaster to switch from one encoding to another, different transcoding profiles must be configured first. These transcoding profiles include the transcoding settings that can be used for the transcoding of streams.

Supported Video and Audio formats include:

- Video Decoding – MPEG2, H.264/AVC , H.265/HEVC, Raw
- Video Encoding – MPEG2, H.264/AVC , H.265/HEVC, Raw
- Audio Decoding – AAC, MPEG-1 Audio Layer I / II / III, MPEG-2 Audio Layer I / II / III, AC-3 (Dolby Digital), Opus



For AC-3, which can contain up to six discrete channels of sound, the transcoder only supports the left and right channels and will ignore the other channels.

- Audio Encoding – AAC, AAC-HE (High Efficiency) and AAC-HEv2; Pass-through (no encoding) – any audio codec (MPEG1/2 / AAC / AC-3, E-AC-3 (Dolby Digital Plus), Opus.



AAC-HE and AAC-HEv2 are only supported for Zixi Broadcasters, v13 and up.

## Transcoder

### Creating New Profiles

The transcoder comes with the following pre-set common video profiles:

1080i60 / 1080i59.94, 1080p30 / 1080p25, 720p60, 720p50 / 720p30, 576i60 / 576p25, 480i60 / 480p29.97

You can also define your own custom profiles.



For downstream satellite workflows, you will probably need to generate constant bitrate (CBR) streams. This setting is configured at the individual input level (not the transcoding profile level). To set a CBR, enter the desired Transport Stream Bitrate (Kbps) as described in the **Transcoding an Input Stream** section.

#### ➔ To Create a New Profile:

1. In the **Transcoder** page, click [+ New Profile](#).

The **Add New Profile** window is displayed:

**Add New Profile**

**Codec type**

H.265  H.264  MPEG2  Raw Video  AAC  Opus  Raw Audio

**Profile configuration**

Preset: Custom

Name:

Encoder: x264

Encoding Profile: Baseline

Resolution:  X

Keep Source Aspect Ratio

Frames per Second: Original

Bitrate Mode: VBR

Average Bitrate (kbps):

Max Bitrate (kbps):

Performance: Very Fast

**Advanced options**

Copy GOP from source:

GOP [frames]:

Closed:  Fixed:

B-Frames: 0

Frame Type: Progressive

Reference Frames:

HRD buffer length (seconds):

CRF:


Max QP:

Apply Close

2. Select the desired **Codec type** - MPEG2, H.264 or H.265, or Raw Video for a video profile or select AAC, Opus, or Raw Audio for an audio profile.
3. Enter the desired fields in the **Profile configuration** (see table below for additional parameters). Select a specific **Preset** profile to automatically fill in the default settings for that profile, OR select **Custom** to create a new profile from scratch.

## Transcoder

### Creating New Profiles

4.  You can edit specific fields of a preset configuration while creating the profile.
5. If you want to configure the **Advanced options** for the video profiles, select **Advanced options** and fill in the respective fields (see table below for additional parameters).
6. Click **Apply**.  
The newly configured profile is created and appears on the screen.

Parameter	Description
Profile configuration	For Video - MPEG2, H.264, H.265, Raw Video
Preset	Select one of the pre-configured presets or select <b>Custom</b> to create a new profile from scratch. Even if you select a preset configuration, you can still change its settings. Default: <b>Custom</b>
Name	Enter a unique name for the profile.
Encoder	Select one of the following options: <ul style="list-style-type: none"><li>• X264 – software-based encoder (available for H.264 only).</li><li>• Intel – Intel-based hardware encoder.</li><li>• NVIDIA – NVIDIA-based hardware encoder.</li></ul>

## Transcoder

### Creating New Profiles

---

---

	Select the relevant encoding profiles: <b>For H.265 video profiles with hardware acceleration:</b> <ul style="list-style-type: none"><li>• Main (default) – this profile supports 8 Bit Per Pixel encoding (BPP).</li><li>• Main 10 – this profile supports 10 Bit Per Pixel encoding (BPP).</li><li>• Main SP - The Main Still Picture profile allows for a single still picture to be encoded with the same constraints as the Main profile. As a subset of the Main profile the Main Still Picture profile allows for a bit depth of 8-bits per sample.</li></ul> <b>For H.264 video profiles:</b> <ul style="list-style-type: none"><li>• High - The primary profile for broadcast and disc storage applications, particularly for high-definition television applications.</li><li>• Main - This profile is used for standard-definition digital TV broadcasts that use the MPEG-4 format as defined in the DVB standard.</li><li>• Baseline - Primarily for low-cost applications that require additional data loss robustness, this profile is used in some video conferencing and mobile applications.</li></ul> <b>For MPEG2 video profiles:</b> <ul style="list-style-type: none"><li>• High – supports SNR and Spatial Scalability and Intra DC Precision of 8, 9, 10, and 11.</li><li>• Main - supports I and P and B Picture Coding Types and Intra DC Precision of 8, 9, 10.</li><li>• Simple – supports I and P Picture Coding Types and Intra DC Precision of 8, 9, 10.</li></ul>
<b>Encoding Profile</b>	
<b>Resolution</b>	Enter the new resolution in pixels, if resizing the video.
<b>Keep Source Aspect Ratio</b>	Select this checkbox (default) to maintain the original aspect ratio.
<b>Frames per Second</b>	Select a new frame rate or select <b>Original</b> to use the original frame rate of the stream. Default: Original Note: When a new frame rate is specified, it is applied using "Smooth frame rate conversion". Note: When the frame rate is converted, captions in the elementary stream are preserved.
<b>Average Bitrate (kbps)</b>	Specify the average bitrate for the profile.
<b>Max Bitrate (kbps)</b>	Specify the maximum bitrate for the profile. Available only if the Bitrate Mode is set to VBR.

---

---



## Transcoder

---

### Creating New Profiles

---

<b>Performance</b>	Select the performance of the transcoding. Possible values range from <i>Ultra Fast</i> which provides the best quality while utilizing high CPU/GPU to <i>Slower</i> which has the lowest quality but uses much less CPU/GPU. Default: <i>Very Fast</i>
<b>Advanced Options</b>	
<b>Copy GOP from source</b>	Select this checkbox if you want to maintain the source GOP length (not structure). Enabling this option will generate an I-frame in the transcoded stream with the exact timestamp as in the source stream. The <b>Copy GOP from source</b> option is only available when selecting the <b>Original</b> option in the <b>Frames per Second</b> field. If you want to change the source GOP length specify the new GOP length in the <b>GOP [Frames]</b> parameter.
<b>GOP [frames]</b>	Specify the number of frames in a <b>GOP</b> (Group of Pictures). Default:60
<b>GOP Settings</b>	<ul style="list-style-type: none"><li>• Closed – for a GOP in which the previous pictures do not need data from the subsequent GOP for bidirectional coding, select the <b>Closed</b> checkbox (default).</li><li>• Fixed – to always use the specified number of frames in the GOP (as opposed to allowing the encoder to override), select the <b>Fixed</b> checkbox (default).</li></ul>
<b>B-Frames</b>	Specify the maximal number of consecutive B frames in the <b>GOP</b> . Default: 0
<b>Frame Type</b>	Specify the frame type – progressive or interlaced. Default: Progressive
<b>Reference Frames</b>	Specify the number of frames that can be referenced for each frame. Default: 2
<b>HRD buffer length (seconds)</b>	Specify the length of the HRD buffer in MPEG-2.
<b>CRF</b>	The Constant Rate Factor (CRF) is the default quality (and rate control) setting for x264 and x265 encoders. You can set the values between 0 and 51, where lower values would result in better quality, at the expense of higher file sizes. Higher values mean more compression, but at some point you will notice the quality degradation.

## Transcoder


### Creating New Profiles

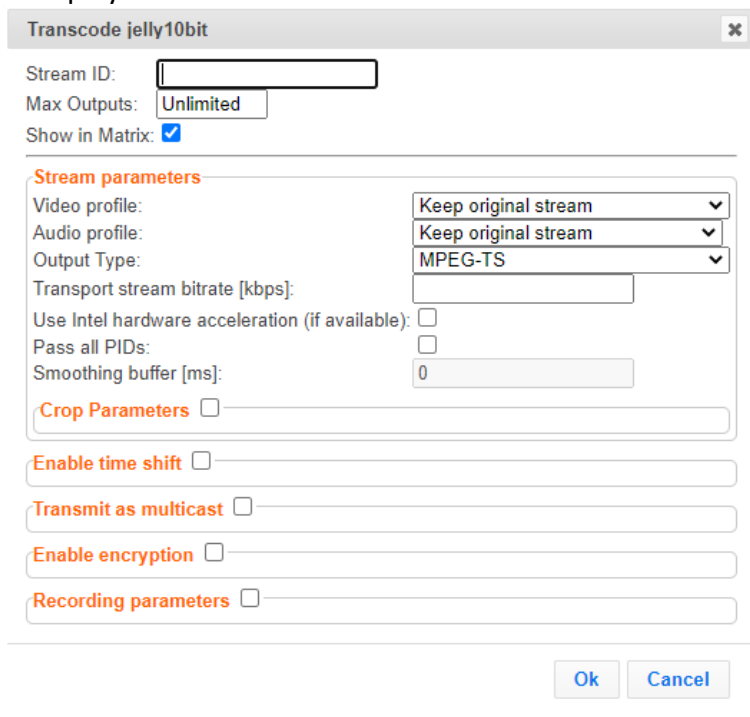
<b>MAX QP</b>	The QP (Quantization) parameter controls how the compression rate. Ranging from 0 (No compression) and 51 (Max compression). Max QP - controls the maximum value the encoder can set for QP. For example, a low number will result in a lower compression rate causing the video to look better, however requiring a higher bitrate. A higher max QP value may reduce the visual quality but results in a lower bitrate.
<b>Bit Depth</b>	<p>Conversion of the 10-bit HDR color encoding source to 8 BPP.</p> <p>This feature is enabled for the following configurations:</p> <ul style="list-style-type: none"><li>• H.264 – Encoder: X264; Encoding Profile: High 10.</li><li>• H.265 – Encoder: Nvidia; Encoding Profile: Main 10.</li></ul> <p>If the source is 10-bit, select one of the following:</p> <ul style="list-style-type: none"><li>• <b>As Source</b> – to maintain a 10-bit color encoding.</li><li>• <b>8 BPP</b> – to converted down/encod to 8-bit color.</li></ul>
<b>Profile configuration</b>	<b>For Audio – AAC, OPUS, Raw Audio.</b>
<b>Name</b>	Enter a unique name for the profile.
<b>Bitrate</b>	Specify the bitrate for the profile.
<b>Sampling Rate</b>	<p>You can convert the audio sampling rate from 44.1kHz to 48kHz and vice versa. Select one of the following options:</p> <ul style="list-style-type: none"><li>• Copy from source – if you do not wish to make any changes.</li><li>• 44.1kHz – to convert a 48kHz audio stream to 44.1kHz.</li><li>• 48kHz – to convert a 44.1kHz audio stream to 48kHz.</li></ul>
<b>Encoding Profile (for AAC)</b>	<ul style="list-style-type: none"><li>• AAC-LC (default) - AAC Low Complexity. The simplest and most widely used and supported profile.</li><li>• HE-AACv1 - AAC High Efficiency 1. This profile uses spectral band replication (SBR) to enhance the compression efficiency in the frequency domain.</li><li>• HE-AACv2 - AAC High Efficiency 2. couples SBR with Parametric Stereo (PS) to enhance the compression efficiency of stereo signals. It is a standardized and improved version of the eAAC+ codec.</li></ul>

## Transcoding an Input Stream

Once you have created the profile, you can use it to transcode an input stream.

### ➔ To Transcode an Input Stream:

1. On the **Inputs** page of Zixi Broadcaster, click the desired row of the input stream that you want to transcode.
2. Click the  button on the right end of the selected row and then **Transcode**. The **Transcode <input stream name>** window with the settings of the selected input stream is displayed.



- 3.
4. Enter the unique **Stream ID** for the new transcoded input stream.
5. Under **Stream Parameters**, in the Video profile and Audio profile fields, do one of the following:
  - Select the desired Video Profile and/or Audio Profile that you have configured.
  - Select **Keep original stream** - if you have not configured a video or audio profile or do not want it transcoded or you want to set specific transcoding parameters for the specific stream. In this case the video or audio input will not be transcoded.
  - Select **Remove stream** – you can use the transcoder to remove the video or audio track from the stream.
6. Fill in additional fields (see table below for additional parameters).
7. Click **OK**.

## Transcoder

---

### Transcoding an Input Stream

The transcoded input stream is created and appears in an embedded window labeled **Transcoded sources** under the original input stream from which it was created.

---

Parameter	Description
<b>Stream Parameters</b>	
<b>Video Profile</b>	Select an existing profile, select <b>Keep original stream</b> if you do not wish to transcode the video stream, or select <b>Remove stream</b> if you want to remove the video track.
<b>Audio Profile</b>	Select an existing profile, select <b>Keep original stream</b> if you do not wish to transcode the audio stream, or select <b>Remove stream</b> if you want to remove the audio track.
<b>Output Type</b>	Select one of the following output types. The outputs are configured as part of the transcoding process, so there is no need to create the Output from the Zixi Broadcaster Outputs screen: <ul style="list-style-type: none"><li>• MPEG-TS – the output will be available as MPEG Transport Stream.</li><li>• NDI – the output will be available as NDI. When selecting NDI output, you can select the following video/audio profiles:<ul style="list-style-type: none"><li>○ Video profile – remove stream or a Raw profile.</li><li>○ Audio profile - remove stream or a Raw profile.</li></ul></li><li>• WebRTC – the output will be available as WebRTC. When selecting NDI output, you can select the following video/audio profiles:<ul style="list-style-type: none"><li>○ Video profile – remove stream, keep original stream, Raw video profile, or if the input stream is not WebRTC compliant, a transcode to WebRTC compliant profile (e.g. H.264 codec with Baseline or Main profile and no B frames).</li><li>○ Audio profile - remove stream, keep original stream, Raw audio profile, or if the input stream is non-Opus, transcode to Opus.</li></ul></li></ul>
<b>Transport Stream Bitrate (Kbps)</b>	Specify the bitrate of the transport stream. This parameter is optional. If you do not specify a TS Bitrate, the stream will be VBR. If you specify a bitrate, the stream will be CBR. Note – CBR is recommended for downstream satellite workflows.
<b>Use Hardware Acceleration (if available)</b>	Select this checkbox to use hardware acceleration component (e.g. NVIDIA or Intel) if it has been installed.
<b>Pass all PIDs</b>	Select this checkbox to pass-through all non-transcoded PIDs.
<b>Smoothing Buffer (ms)</b>	Specify how much of the transcoded stream (in milliseconds) to store before transmitting it back to the server in order to prevent bursts. If the <b>Transport stream bitrate [kbps]</b> field is empty, the smoothing buffer is disabled.

---

## Transcoder

### Transcoding an Input Stream

Parameter	Description
	If the <b>Transport stream bitrate [kbps]</b> field is NOT empty, it is recommended to enter around 200ms, which should be enough to prevent bursts.
<b>Crop Parameters</b>	Select this checkbox to set the cropping parameters.
<b>Cropping Mode</b>	Set the units used for defining the crop. Options are: <i>Pixels</i> or <i>Percentages</i> . Note: Depending on the type of content, in some cases using different crop values for low resolution outputs might be appropriate. For example, if the content includes both video content and textual content, the text would not be readable at lower resolutions and thus might be better to exclude. Another use case for using different crop values for each output would be to slice a 360 or VR input video.
<b>Left, Top</b>	Specify the left and top positions for the crop.
<b>Width x Height</b>	Specify the width and height values for the crop.
<b>Enable time shift</b>	
<b>Maximum delay [hh:mm:ss]</b>	Specify the time of the delayed transmission of this stream. Maximum: 24 hours.
<b>Transmit as multicast</b>	Selecting this checkbox enables the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling Multicast Pull in the Settings. This setting requires enabling Multicast Pool (Settings > Multicast Pool).
<b>Multicast Only</b>	Selecting this checkbox will force the transmission of this stream only in multicast.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a> .
<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).
<b>Encryption key (for encryption enabled)</b>	Click <b>Generate</b> to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.
<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.
<b>Destination Type</b>	Select the radio button for the desired storage type and then fill in the relevant parameters.

## Transcoder

---

### Transcoding an Input Stream

---

Parameter	Description
	<ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk.</li><li>• <b>Record to S3</b> – the file is saved to your AWS S3 Bucket.</li></ul>
<b>Record to Disk Parameters</b>	
<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours
<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.
<b>Record to S3 Parameters</b>	
<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p>

## Transcoder

### Transcoding Templates

Parameter	Description
	<ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>URL</b>	Specify the URL of the S3 bucket.
<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.


## Transcoding Templates

Instead of manually selecting input streams to be transcoded, you can define templates with rules for automatic transcoding of input streams according to the settings of a pre-defined Transcoding Template. Every time a new input stream, which matches the conditions of the template, is added, it will be automatically transcoded into the transcoded streams defined in the template's settings. The input streams are identified by system according to naming patterns.



The transcoding profiles used in the transcoding templates have to be created in advance (for more information see [Creating New Profiles](#)).

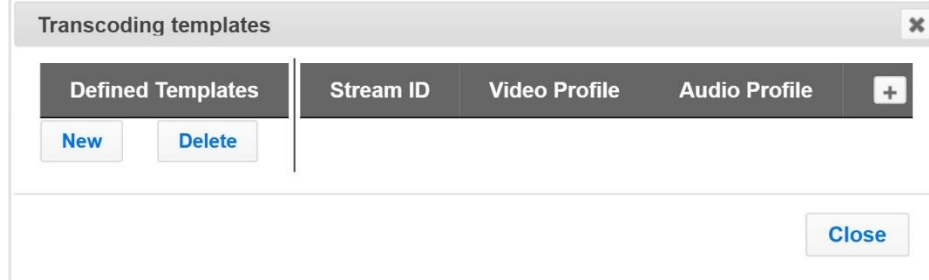
#### ➔ To create a new transcoding template:

1. Click the **Inputs** tab at the top of the Zixi Broadcaster administrative screen.
2. In the Menu bar, click the Templates (  ) button.

## Transcoder

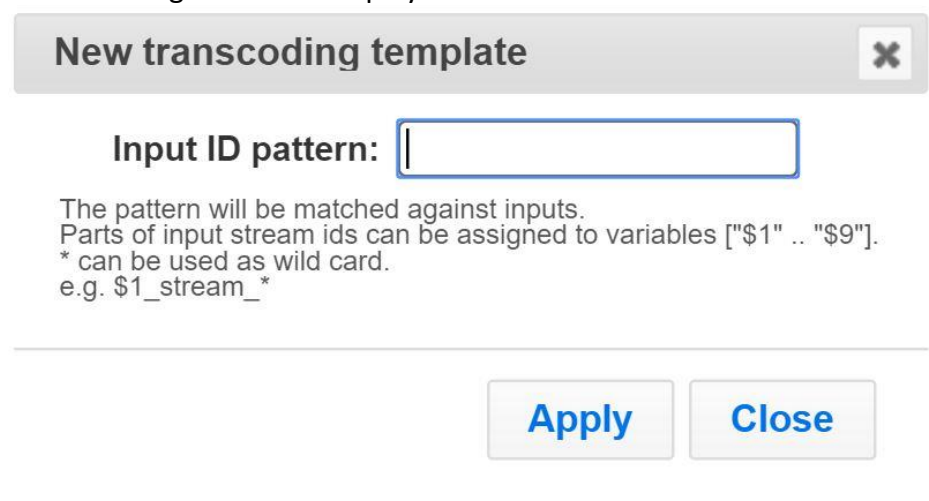
### Transcoding Templates

The **Transcoding Templates** window is displayed:



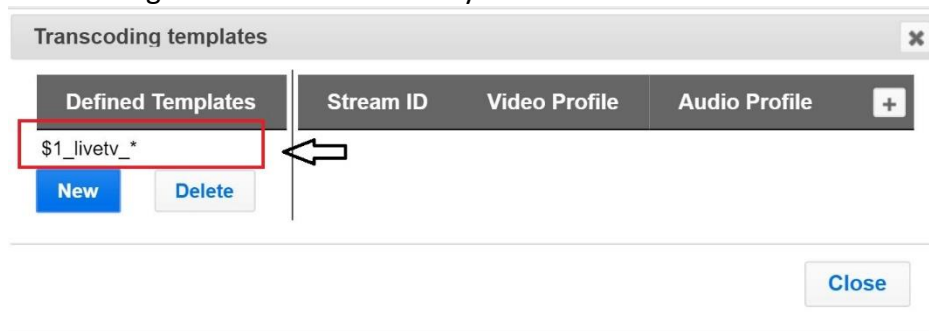
3. Click **New** to define a new template.

The following window is displayed:



4. In the **Input ID pattern** field, enter a pattern for the Stream ID of the input stream that will be automatically transcoded. This pattern will be used to identify the input stream and can include variables that are marked with the "\$" sign followed by a number (e.g. \$1, \$2, etc.) and/or with an asterisk "\*" wildcard. The patterns should also include a part of the string that requires an exact match. For example, the pattern "\$1\_stream\_\$2kbps" will match an input stream "talk\_show\_stream\_600kbps".
5. Click **Apply**.

The new template is added to the list of transcoding templates. However, the transcoding will not be active until you define transcoded stream rules.



6. Select the newly created template and click the **+** button, located at the right-hand corner to define a transcoded stream rule.



The **Transcode <template\_name>** window opens:

7. Enter the unique **Stream ID**. for the transcoded stream.
8. In the **Max Outputs** field, define the maximal number of outputs that can be connected to the stream.
9. Under **Stream Parameters**, select the desired Video Profile and/or Audio Profile that you have configured. If you have not configured a video or audio profile you can select **Keep original stream**. In this case the video or audio input will not be transcoded. You can also set specific transcoding parameters for the specific stream (see table below for more details).
10. Click **OK**.

The transcoded input stream rule is created and appears as follows:

Defined Templates	Stream ID	Video Profile	Audio Profile
\$1_livetv_*	\$1_livetv_	trans1	Original stream

## Transcoder

### Transcoding Templates

If an input stream enters Zixi Broadcaster and matches the **Input ID pattern**, it will be transcoded according to all the transcoded input stream rules that were created.

Parameter	Description
<b>Video Profile</b>	Select an existing profile or select <b>Keep original stream</b> if you do not wish to transcode the video stream.
<b>Audio Profile</b>	Select an existing profile or select <b>Keep original stream</b> if you do not wish to transcode the audio stream.
<b>Transport Stream Bitrate (Kbps)</b>	Specify the bitrate of the transport stream. This parameter is optional. If you do not specify a TS Bitrate, the stream will be VBR. If you specify a bitrate, the stream will be CBR.
<b>Use Hardware Acceleration (if available)</b>	Select this checkbox to use hardware acceleration component if it has been installed.
<b>Pass all PIDs</b>	Select this checkbox to pass-through all non-transcoded PIDs.
<b>Enable time shift</b>	<b>Select to enable delayed transmission.</b>
<b>Maximum delay [hh:mm:ss]</b>	Specify the time of the delayed transmission of this stream. Maximum: 24 hours.
<b>Transmit as multicast</b>	Selecting this checkbox enables the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling Multicast Pull in the Settings. This setting requires enabling Multicast Pool (Settings > Multicast Pool).
<b>Multicast Only</b>	Selecting this checkbox will force the transmission of this stream only in multicast.
<b>Recording Parameters</b>	
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours

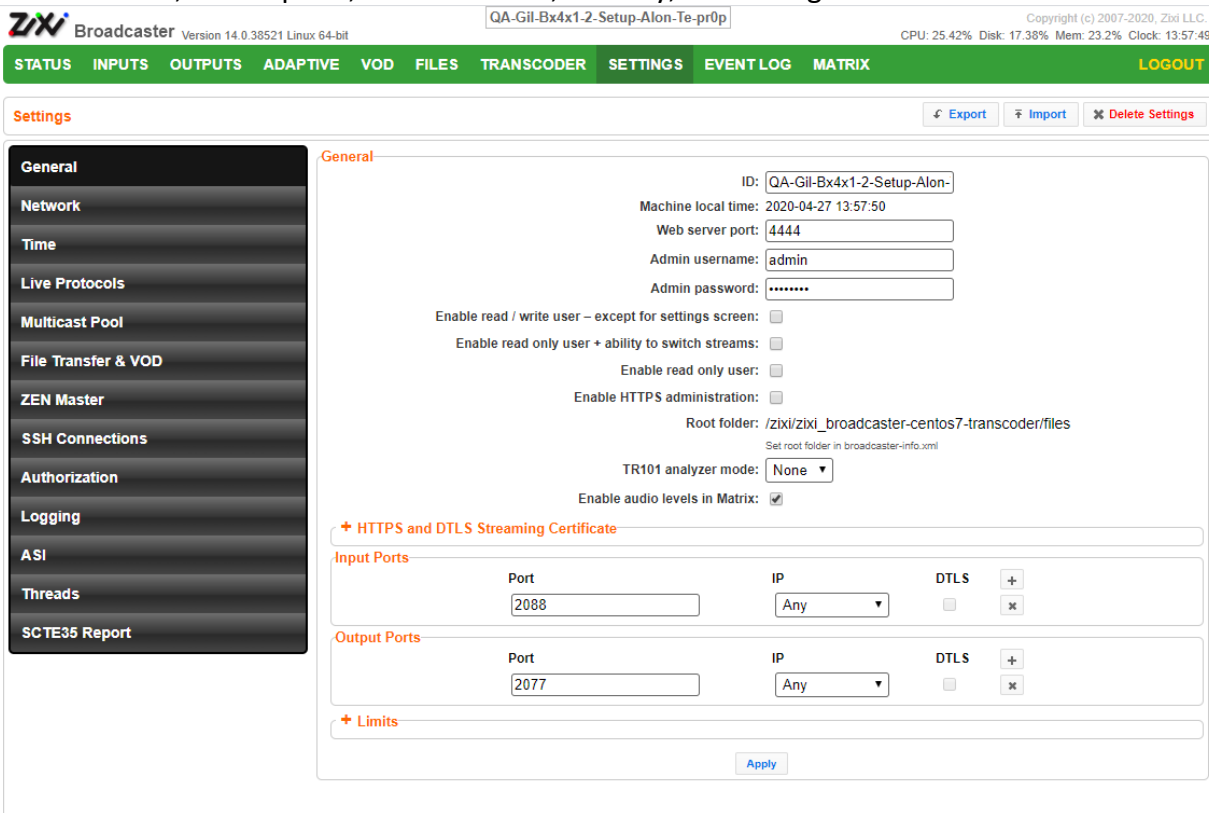
11. Repeat steps 6-10 above to create additional transcoded input stream rules.

# Server Settings

Zixi Broadcaster is pre-configured with default settings, so generally there is no need to change the server settings. However, some users may want to change specific settings to meet their specific needs.

## General

The General Settings screen includes basic configuration parameters for the server including the server’s ID, access ports, authorizations, security, and storage.



Field	Description
<b>General</b>	
ID	The unique (alphanumeric) name of the Zixi Broadcaster server. Typically: host name. This field is editable.
Machine local time	Shows the current time for the machine.

## Server Settings

---

### General

---

<b>Web Server Port</b>	The port number of the web server administration. You can configure this parameter to any legal port number which is supported by your Firewall. Default: 4444
<b>Admin User Name</b>	The user name of the administrator. The administrator is authorized to modify all parameters in the system as well as manage all the outputs and inputs on the server.
<b>Admin Password</b>	The password of the administrator.
<b>Enable read / write user - except for settings screen</b>	Select this checkbox to enable an "Operator" user. This type of user can read and write throughout all of the screens, except for the screens under the Settings menu. You can modify the user name (i.e. enter another name instead of Operator) and password.
<b>Enable read-only user +ability to switch streams</b>	Select this checkbox to enable a "User" user. This type of user can read throughout all of the screens as well as the ability to switch between existing streams. You can modify the user name (i.e. enter another name instead of User) and password.
<b>Username</b>	The account Username.
<b>Password</b>	The account password.
<b>Enable read only user</b>	Select this checkbox to enable an "Observer" user. This type of user can read throughout all of the screens. You can modify the user name (i.e. enter another name instead of Observer) and password.
<b>Enable HTTPS administration</b>	Enables secure HTTPS connection to the server for administration. Enabling HTTPS requires the relevant certificates and private keys. To enable HTTPS, click on the checkbox and upload the required files.
<b>HTTPS certificate uploaded (for HTTPS enabled)</b>	If no certificate has been uploaded, click <b>Upload</b> and navigate to the file location.
<b>HTTPS private key uploaded (for HTTPS enabled)</b>	If no private key has been uploaded, click <b>Upload</b> and navigate to the file location.
<b>HTTPS private key passphrase (for HTTPS enabled)</b>	Enter the Private key passphrase for your HTTPS account.
<b>SSL Setup Status</b>	Displays the status of the SSL certificate upload process. Incomplete – the SSL setup process is incomplete.

---

## Server Settings

---

### General

---

---

	OK – the SSL setup process is complete.
<b>Root folder</b>	Displays the configured root folder. You can modify the root folder by modifying the broadcaster-info.xml, which is located in the Zixi Broadcaster installation folder (<Broadcaster_Installation_Folder>\Zixi\Broadcaster\)
<b>TR101 analyzer mode</b>	Select the mode in which the TR101 analyzer will operate: None – no TR101 analysis ATSC – based on the Advanced Television Systems Committee recommendations. DVB – based on the Digital Video Broadcasting specifications.
<b>Enable audio levels in matrix</b>	Select this option to enable audio in Matrix view. When not selected the audio will be turned off.

---

### HTTPS and DTLS Streaming Certificate

---

**Certificate uploaded (for DTLS enabled)** If no certificate has been uploaded, click Upload and navigate to the file location.  
**Note:** This is the same certificate as the HTTPS certificate.

---

**Private key uploaded (for DTLS enabled)** If no private key has been uploaded, click Upload and navigate to the file location.

---

**Private key passphrase (for DTLS enabled)** Enter the Private key passphrase for your DTLS account.

---

### Input ports

---

**Port** The port that is used for Input streams. You can modify the port number and/or add additional ports by adding a “,” between them.  
Adding additional ports may be necessary when there is a chance that the ISP will block a certain port. Click '+' to add additional ports.  
Default: 2088

---

**IP** Enter an IP address. The default enables **Any** IP address.

---

**DTLS** Click the DTLS check box to activate DTLS automatic encryption of the streams.

---

**+ (add ports)** Click + to add additional ports.  
**Note:** For each port you can select whether or not to activate DTLS encryption. This enables streaming of non-encrypted streams on one port and encrypted streams on a separate port.

---

### Output ports

---

## Server Settings

---

### General

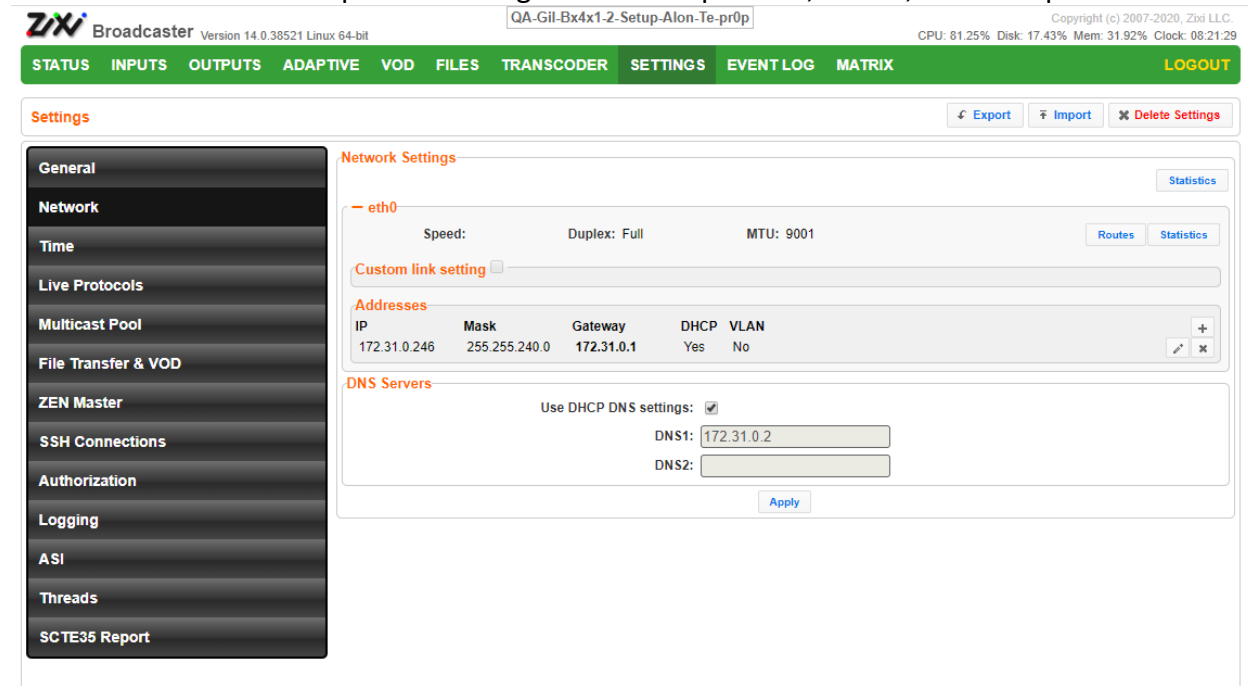
---

<b>Port</b>	The port that is used for the Output streams. Displays the Output port(s). You can modify the port number and/or add additional ports by adding a “,” between them. Adding additional ports may be necessary when there is a chance that the ISP will block a certain port. Click '+' to add additional ports. Default: 2077
<b>IP</b>	Enter an IP address. The default enables <b>Any</b> IP address.
<b>DTLS</b>	Click the DTLS check box to activate DTLS automatic encryption of the streams.
<b>+ (add ports)</b>	Click + to add additional ports. <b>Note:</b> For each port you can select whether or not to activate DTLS encryption. This enables streaming of non-encrypted streams on one port and encrypted streams on a separate port.
<b>Limits</b>	Select to configure thresholds for generating an alarm.
<b>CPU [%]</b>	The CPU utilization threshold. If this limit is exceeded, the CPU indicator at the top of the screen will appear in red, and an alarm will be generated in the event log. Default: 90
<b>Memory [%]</b>	The memory utilization threshold. If this limit is exceeded, the Memory indicator at the top of the screen will appear in red, and an alarm will be generated in the event log Default: 90
<b>GPU [%]</b>	The GPU utilization threshold. If this limit is exceeded, the GPU indicator at the top of the screen will appear in red, and an alarm will be generated in the event log. Default: 0
<b>Input Bandwidth [kbps]</b>	Bandwidth utilization threshold for input streams. Default: Unlimited.
<b>Output Bandwidth [kbps]</b>	Bandwidth utilization threshold for output streams. Default: Unlimited.

---

# Network

The Network screen enables the configuration of the network settings of the Zixi Broadcaster. The screen includes the option to configuration multiple NICs, VLANs, and multiple DNS Servers.




Field	Description
<b>Network Settings</b>	Click to configure the specified NIC. Displays the following parameters: <ul style="list-style-type: none"> <li>Speed – the network card speed</li> <li>Duplex – whether the network card is full-duplex or half-duplex</li> <li>MTU - the size of the largest protocol data unit (PDU) that can be communicated in a single network layer transaction.</li> </ul>
<b>eth0</b>	
<b>Custom link settings</b>	Select this checkbox to configure different link settings than the default settings.
<b>Addresses</b>	Displays the current IP address that is configured for the link. You edit these settings by clicking the Edit (pencil icon) button. Click the '+' button to add another link.
<b>DNS Servers</b>	
<b>Use DHCP DNS settings</b>	Select this option to use the DHCP settings instead of the DNS. If not selected, configure at least one DNS server.

## Server Settings

---

### Network

#### ➔ To Display Network Statistics:

1. From the Settings page of the Zixi Broadcaster, click **Network** on the left-hand side. The various Network settings are displayed.
2. Click  on the right-hand side. The **UDP statistics** window is displayed.

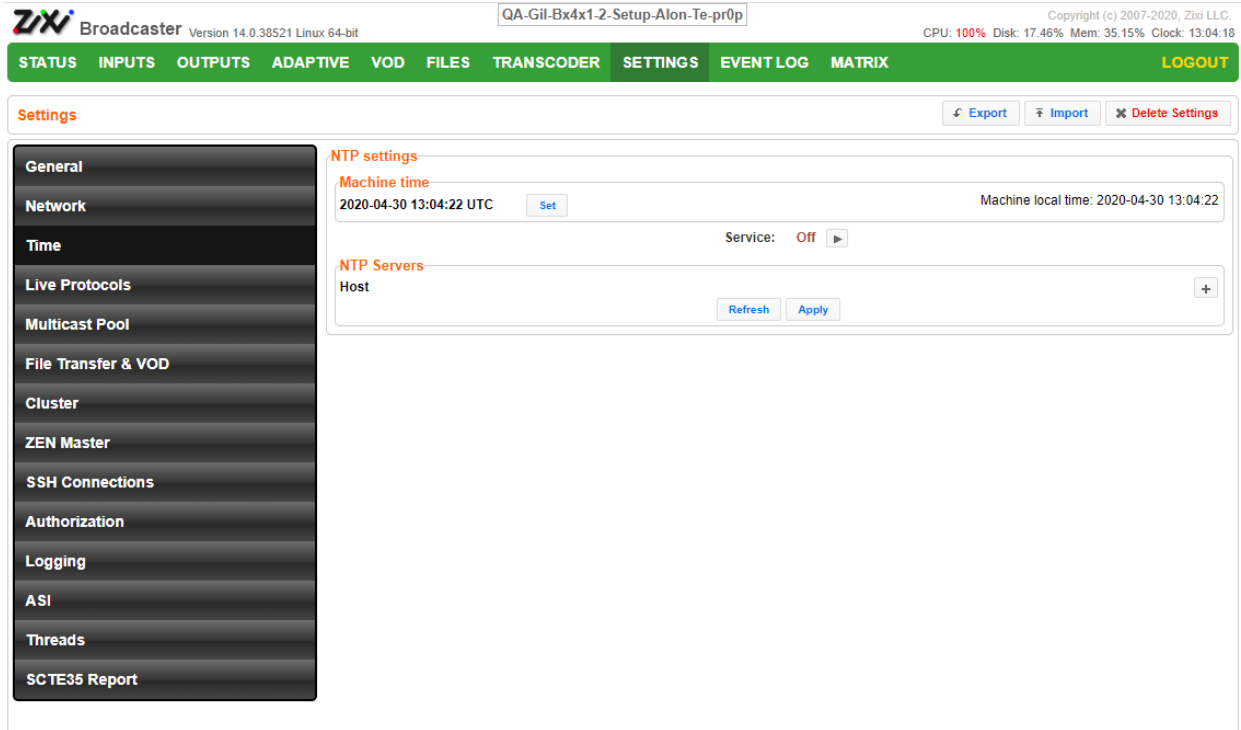
Field	Description
####Packets received	Number of packets on network received.
####packets to unknown port received	Number of packets to unknown port on network received.
####packet receive errors	Number of packets on network which received errors
####packets sent	Number of packets on network sent.
####SndbufErrors	Number of Sent buffer errors

Click the **Refresh** button to update the data or **Close** to close the window



# Time

The Time screen enables the configuration of the system time synchronization for the system. The time can be synchronized with the machine local time or to an external NTP server.



Field	Description
<b>NTP Settings</b>	
<b>Machine Time</b>	Displays the current machine local time. To set the time click <b>Set</b> and modify the values.
<b>Service</b>	
<b>NTP Servers</b>	Configure external NTP servers for time synchronization. Click the + button to add an NTP server and specify the Host IP address.

## Server Settings

### Live Protocols

## Live Protocols

The Live Protocols screen enables the configuration of the HTTP/HTTPS/RTMP/WebSocket server that will broadcast the streams using live streaming protocols, particularly for adaptive streaming and internet protocols.

The screenshot shows the Zixi Broadcaster interface. At the top, there's a navigation bar with tabs: STATUS, INPUTS, OUTPUTS, ADAPTIVE, VOD, FILES, TRANSCODER, SETTINGS (selected), EVENT LOG, MATRIX, and LOGOUT. Below the navigation bar, there's a 'Settings' section with a sidebar on the left containing menu items: General, Network, Time, Live Protocols (highlighted), Multicast Pool, File Transfer & VOD, Cluster, ZEN Master, SSH Connections, Authorization, Logging, ASI, Threads, and SCTE35 Report. The main content area is titled 'Settings' and has buttons for 'Export', 'Import', and 'Delete Settings'. It is divided into two sections: 'HTTP Server' and 'RTMP Server'. The 'HTTP Server' section includes fields for 'HTTP/HTTPS server public IP', 'Enable HTTP server' (checked), 'HTTP server port' (7777), 'Enable HTTPS server' (checked), 'HTTPS server port' (443), 'HTTPS certificate uploaded' (No) with an 'Upload' button, 'HTTPS private key uploaded' (No) with an 'Upload' button, 'HTTPS private key passphrase' (masked), 'SSL setup status' (incomplete), 'Enable WebSocket server' (checked), 'WebSocket server port' (8100), 'Enable FLV' (checked), 'Enable HLS' (checked), 'Enable CMAF(DASH + IMP4 HLS)' (checked), 'Enable SHOUTcast' (checked), 'Enable HLS Fast playback' (checked), 'Sub gop HLS Fast playback' (unchecked), 'HLS DVR maximum recording duration [HH:MM]' (24:00), 'Allow Automatic HTTP Push input' (unchecked), and 'Allow Automatic HTTP Pull output' (unchecked). An 'Apply' button is at the bottom of this section. The 'RTMP Server' section includes 'Enable RTMP Server' (checked), 'RTMP server port' (1935), 'Allow automatic RTMP input' (unchecked), and 'Allow automatic RTMP output' (unchecked). An 'Apply' button is at the bottom of this section.

Field	Description
HTTP Server	
HTTP/HTTPS Server Public IP	Override the Zixi IP (e.g. download.Zixi.com) with a public IP.
Enable HTTP Server	Enable a non-secure connection over HTTP.
HTTP Server Port	Defines the HTTP Server Port. Default:7777
Enable HTTPS Server	Enable a secure connection over HTTPS.

## Server Settings

---

### Live Protocols

---

---

**HTTPS Server Port** Defined the HTTPS server port. Default: 443

---

**HTTPS certificate uploaded (for HTTPS enabled)** If no certificate has been uploaded, click Upload and navigate to the file location.

---

**HTTPS private key uploaded (for HTTPS enabled)** If no private key has been uploaded, click Upload and navigate to the file location.

---

**HTTPS private key passphrase (for HTTPS enabled)** Enter the Private key passphrase for your HTTPS account.

---

Displays the status of the SSL certificate upload process.

**SSL Setup Status** Incomplete – the SSL setup process is incomplete.

OK – the SSL setup process is complete.

---

**Enable WebSocket server** Select checkbox to enable the WebSocket server. The WebSocket server uses the WebSocket protocol to stream content. The WebSocket protocol provides full-duplex communication channels over a single TCP connection.

---

**WebSocket Server Port** If you have enabled WebSocket, enter a dedicated port for WebSocket. (Default: 8100)

---

**Enable FLV** Enable the streaming via the FLV protocol.

---

**Enable HLS** Enable the streaming via the HLS protocol.

---

**Enable CMAF (DASH + fMP4 HLS)** Enable the streaming via the CMAF protocol.

---

**Enable SHOUTcast** Enable the streaming via the SHOUTcast protocol.

---

**Enable HLS Fast playback** Enables low latency HLS and DASH. When this setting is enabled, all input streams on the Zixi Broadcaster will be published to the origin server as HLS streams, so only enable this feature if it will be actually used.

---

**Sub GOP HLS Fast playback** Select this option to enable publishing of HLS chunk before the response is complete. This means that Zixi Broadcaster will start sending the HTTP response as chunks of data when they are ready before the complete response is available.

---

**HLS DVR maximum recording duration** Defines the maximum recording duration for HLS DVR. Default: 24 hours. [HH:MM]:

---

## Server Settings

---

### Live Protocols

---

---

**Allow Automatic HTTP Push input** Automatically creates an HTTP Push input without the need to define it in the Input screen.

---

---

**Allow Automatic HTTP Pull output** Automatically creates an HTTP Pull output without the need to define it in the Output screen. This works in tandem with Zixi Feeder (Remote HTTP Pull Output setting, which enables sending a Zixi-protected stream to Zixi Broadcaster and then Zixi Broadcaster automatically outputs the stream as HTTP Pull) and SDK (which includes a `zixi_http_pull_out_config` option). By enabling this option, the HTTP Pull streams will be automatically created as they arrive into Zixi Broadcaster and displayed in the Output screen.

---

---

### RTMP Server

---

---

**Enable RTMP Server** Enables Zixi Broadcaster to handle incoming RTMP streams.

---

---

**RTMP Server Port** The port through which the RTMP stream will be transmitted. Default: 1935

---

---

**Allow Automatic RTMP Input** Automatically creates an RTMP input without the need to define it in the Input screen.

---

---

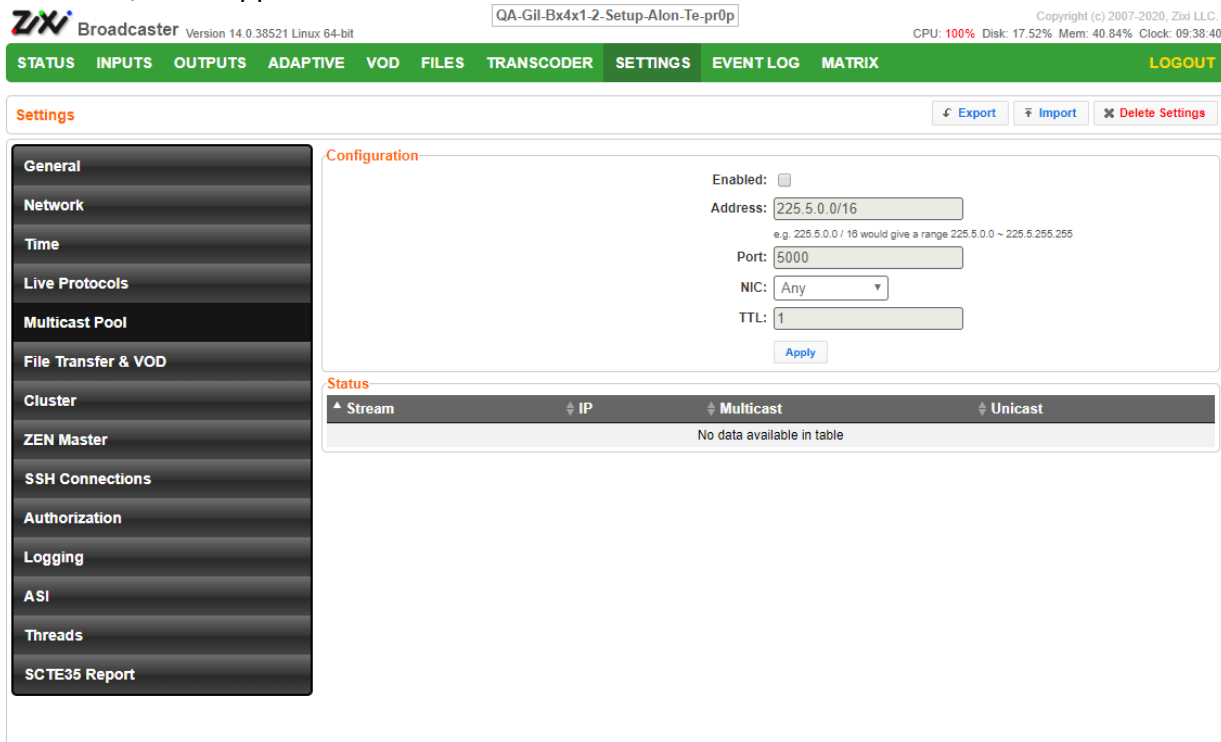
**Allow Automatic RTMP Output** Automatically creates an RTMP output without the need to define it in the Output screen. This works in tandem with Zixi Feeder (**Remote RTMP Output** setting, which enables sending a Zixi-protected stream to Zixi Broadcaster and then Zixi Broadcaster automatically outputs the stream as RTMP) and SDK (which includes a `zixi_rtmp_out_config` option). By enabling this option, the RTMP streams will be automatically created as they arrive into Zixi Broadcaster and displayed in the Output screen.

---

---

### Multicast Pool

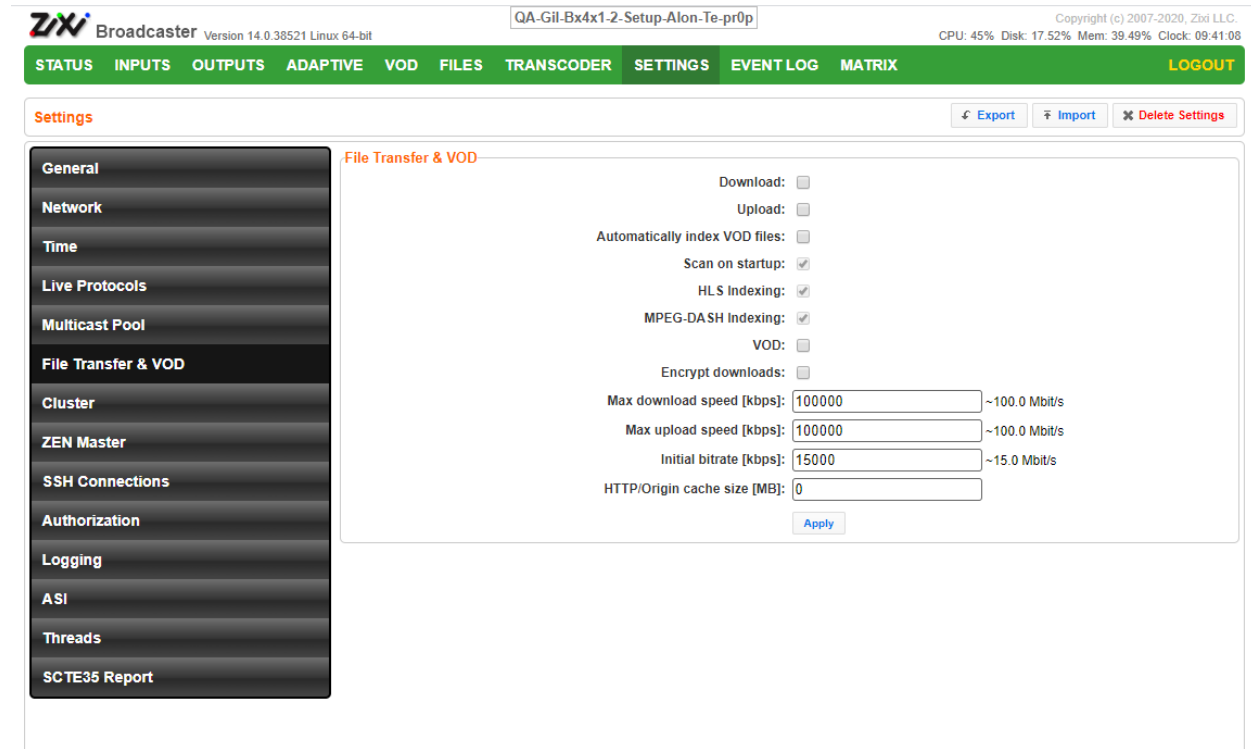
Multicast pool is a collection of multicast IP addresses that automatically assign a multicast IP when an input is being pulled out of the Zixi Broadcaster. When the “multicast pool” setting is enabled, you do not have to define a specific multicast IP address for every push stream. Instead, you simply enable the Multicast Only parameter and Zixi Broadcaster will assign the multicast IP automatically when it pulls the stream out. When the multicast stream is connected, it will appear in the Status table at the bottom of the screen.



Field	Description
<b>Configuration</b>	
Enabled	Enables multicast pool
Address	Specify the range of IP addresses (CIDR format).
Port	Specify the port that will be shared by all of the members of the pool.
NIC	Specify the network card to use for the outgoing multicast traffic.
TTL	Specify the Time To Live.

## File Transfer & VOD

The File Transfer and VOD settings screen enables the download of files and the playback of VOD content and, in addition, specifies the speed limits (download/upload). The settings in this screen are related to the VOD and Files screens.



Field	Description
<b>File Transfer &amp; VOD</b>	
<b>Download</b>	Enables download of files through the Files menu of the Zixi Broadcaster.
<b>Upload</b>	Enables upload of files through the Files menu of the Zixi Broadcaster
<b>Automatically index VOD files</b>	Indexes the VOD files in the various VOD folders. In some cases, the indexing of VOD files can affect performance. Using this control, you can decide when to index the VOD files in order to ensure optimal performance during peak times.
<b>Scan on startup</b>	Indexes the files in the VOD folder that have not been indexed during startup of the Zixi Broadcaster.
<b>HLS indexing</b>	Enables indexing of HLS streams. The indexing creates an M3U8 playlist file.
<b>MPEG-DASH indexing</b>	Enables indexing of MPEG-DASH streams. The indexing creates a .mpd (MPEG-DASH Media Presentation Description) file.

## Server Settings

---

### File Transfer & VOD

---

---

<b>VOD</b>	Enables the VOD feature.
<b>Encrypt downloads</b>	Encrypts files that are downloaded.
<b>Max download speed [kbps]</b>	Limits the amount of bandwidth used to download files
<b>Max upload speed [kbps]</b>	Limits the amount of bandwidth used to upload files
<b>Initial bitrate [kbps]</b>	Limits the amount of bandwidth used during initial download/upload (before cache)
<b>HTTP/Origin cache size [MB]</b>	Limits the amount of cache memory that can be used for cache

---

---

## Cluster

Zixi Broadcaster enables two types of configurations:

- **Cluster configuration** - a cluster of Zixi Broadcaster servers in a LAN for load balancing purposes. The different Zixi Broadcaster servers will communicate with each other and are “aware” of the Input/Output traffic of each node in order to redirect the streams to the least occupied Zixi Broadcaster instance. All Zixi Broadcasters should have a separate **Public IP** (NIC #1) and in addition, they all should have an internal IP on the LAN (**Cluster Internal IP**) to communicate over multicast (NIC #2). When a request received by the Zixi Broadcaster, it accepts it, or transmit it internally to another Zixi Broadcaster that is less loaded and has the requested channel. The Cluster feature is currently supported for:
  - Inputs - Zixi Push inputs
  - Outputs - Zixi Pull outputs, HLS outputs



If you want to use more advanced load balancing and high availability features, instead of configuring the cluster within each Zixi Broadcaster (as described below) it is recommended to configure the cluster within ZEN Master, which requires a single cluster configuration and then automatically manages the load balancing and high-availability, while enabling monitoring, reporting, and other advanced features. For more information, see:

<https://zixidocumentation.atlassian.net/wiki/spaces/ZMUG/pages/388005967/Adding+a+Broadcaster+Cluster+Automatic+Scaling>.

- **Edge-Origin configuration** - the cluster may be used to forward the stream from one Zixi Broadcaster server (origin) to other Zixi Broadcaster servers (edge) which are closer to the user or is located in a less sensitive environment. When the end-user connects to the edge server to view a certain stream, the edge server will retrieve the stream from the origin server and will seamlessly deliver it to the end-user.



# Server Settings

## Cluster

ZIXI Broadcaster Version 14.0.38521 Linux 64-bit QA-Gil-Bx4x1-2-Setup-Alon-Te-pr0p Copyright (c) 2007-2020, Zixi LLC. CPU: 98.09% Disk: 17.52% Mem: 39.33% Clock: 09:47:47

STATUS INPUTS OUTPUTS ADAPTIVE VOD FILES TRANSCODER **SETTINGS** EVENT LOG MATRIX LOGOUT

Settings Export Import Delete Settings

**General**

Network

Time

Live Protocols

Multicast Pool

File Transfer & VOD

**Cluster**

ZEN Master

SSH Connections

Authorization

Logging

ASI

Threads

SCTE35 Report

**Origin**

Enabled:

Host:  Port:

Alternative host:  Port:

HTTP/Origin cache size [MB]:

**Live Streams**

Latency [ms]:

Timeout [s]:

Apply

**Cluster**

Enabled:

Public IP:

Cluster internal IP:  TTL:

Port:

Local IP:

Balance inputs:

Balance outputs:

Apply

Field	Description
<b>Origin</b>	
<b>Enabled</b>	Enables Edge-Origin configuration.
<b>Host</b>	The primary host IP address of the origin server.
<b>Port</b>	The port number in the origin server for transmitting the stream. Default: 2088
<b>Alternative Host</b>	The IP address for a secondary server. If the primary origin server is unavailable, the data will be retrieved from the secondary origin server.
<b>Port</b>	The port number for the secondary origin server. Default: 2088
<b>HTTP/origin cache size [MB]</b>	The size of the cache that will be used to serve multiple users. The edge server will retrieve the data from the origin and cache it for additional users in order to save bandwidth and increase performance.
<b>Live Streams</b>	
<b>Latency [ms]</b>	Specify the buffer size between the edge and the origin.
<b>Timeout [s]</b>	Specify the maximum amount of time in which the edge will attempt to connect to the origin. After the timeout period the edge server will attempt a connection with the secondary origin server.

## Server Settings

---

### Cluster

<b>Cluster</b>	
<b>Enabled</b>	Enables Cluster configuration.
<b>Public IP</b>	The public-facing IP address of this Zixi Broadcaster server for external access. A unique Public IP must be assigned to each broadcaster in the cluster.
<b>Cluster Internal IP</b>	The internal IP (multi-cast) address for communication between the various servers in the cluster. This address is used to communicate load information forwarding the streams within the cluster. The same multicast IP address should be used across all instances in the cluster.
<b>TTL</b>	Specify the Time To Live parameter.
<b>Port</b>	Specify the port that are used in all the participating Broadcaster servers in the cluster.
<b>Local IP</b>	Specify the local IP address which the network employs for internal communication.
<b>Balance Inputs</b>	Enables the balancing of the inputs.
<b>Balance Outputs</b>	Enables the balancing of the outputs.

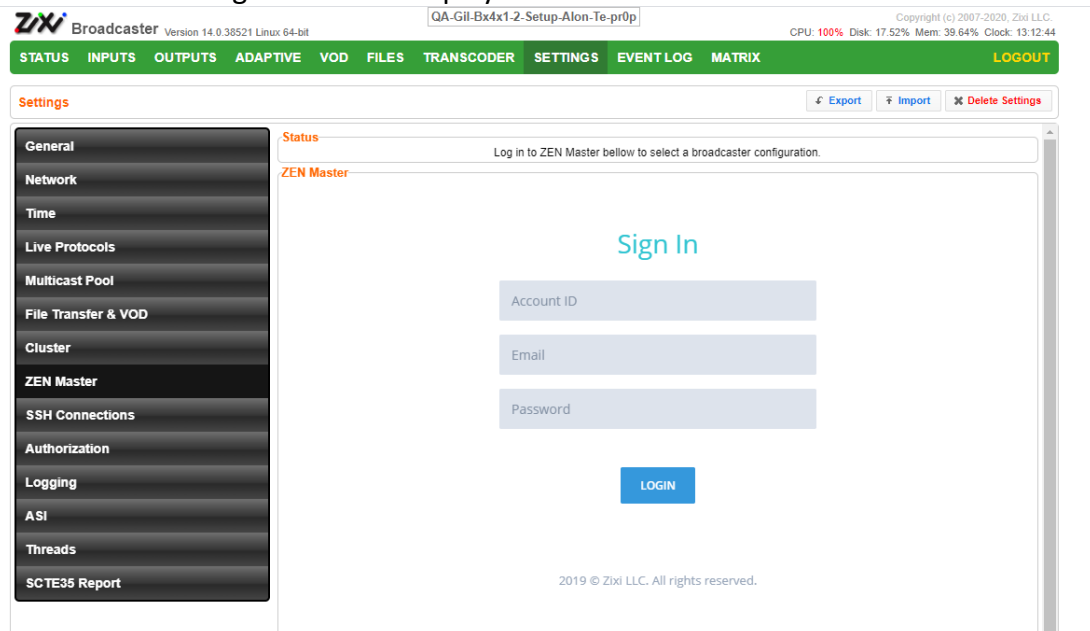
## Zen Master

The ZEN Master screen is used to add the current Zixi Broadcaster to an existing Broadcaster Cluster in Zen Master that has been configured for "Manual Scaling". This process involves pre-configuring a "manual scaling" Broadcaster Cluster in ZEN Master, then, still in ZEN Master, creating a manual Zixi Broadcaster within this cluster, and then finally from the ZEN screen of the current Zixi Broadcaster UI, selecting the previously created Broadcaster Cluster and Zixi Broadcaster (that we had created though the ZEN Master UI). This process will generate a connection between the logical representation of the Zixi Broadcaster in ZEN Master and the Zixi Broadcaster system.

### ➔ To connect the current Zixi Broadcaster to ZEN Master:

1. In **ZEN Master**, create a "manual scaling" Broadcaster Cluster and add a Broadcaster entity to it by following the procedure described in the **Adding a Broadcaster Cluster – Manual Scaling** section in the **ZEN Master User Guide**.
2. In ZEN Master, in the main navigation, click **Settings**.
3. In the **SSH Keys** tab, click **Download** on the relevant SSH Key to download the key to your machine.
4. Login to the **Zixi Broadcaster** UI (it could be a locally installed Zixi Broadcaster, e.g. `http://localhost:4444` or a remote Zixi Broadcaster).  
The Zixi Broadcaster UI opens.
5. Go to **Settings > ZEN Master**.

The ZEN Master login fields are displayed:



The screenshot shows the Zixi Broadcaster Settings page. The top navigation bar includes STATUS, INPUTS, OUTPUTS, ADAPTIVE, VOD, FILES, TRANSCODER, SETTINGS, EVENT LOG, MATRIX, and LOGOUT. The left sidebar lists various settings categories, with ZEN Master selected. The main content area displays the ZEN Master login form, which includes a Status bar, a message to log in to ZEN Master, and three input fields: Account ID, Email, and Password. A LOGIN button is located below the input fields. The footer of the page indicates copyright information for Zixi LLC.

6. In the **Account ID** field, enter the customer domain of your ZEN Master account (e.g. if you login to ZEN at `demo.zen.zixi.com` then your Account ID is `'demo'`).

## Server Settings

### Zen Master



The user cannot be an SSO (Single Sign-On) user. If you are using an SSO account, you will have to create another user to sign in through this screen.

- In the **Email** field, enter the username for your ZEN Master account.
- In the **Password** field, enter the password from your ZEN Master account.
- Click **Login**.  
A list of Broadcasters configured in ZEN Master is displayed.
- For the Broadcaster that you are adding, which should currently be in **Pending** mode, click **Select**.  
Details about the Broadcaster are shown in the **Status**
- Click **Refresh**.  
The connection is configured automatically.
- In **ZEN Master**, in the main navigation, click **Broadcasters**.
- Verify that the status of the newly configured Broadcaster cluster is **OK**.
- Verify that the status of each Broadcaster that was added to the cluster has changed from **Pending** to **OK**.

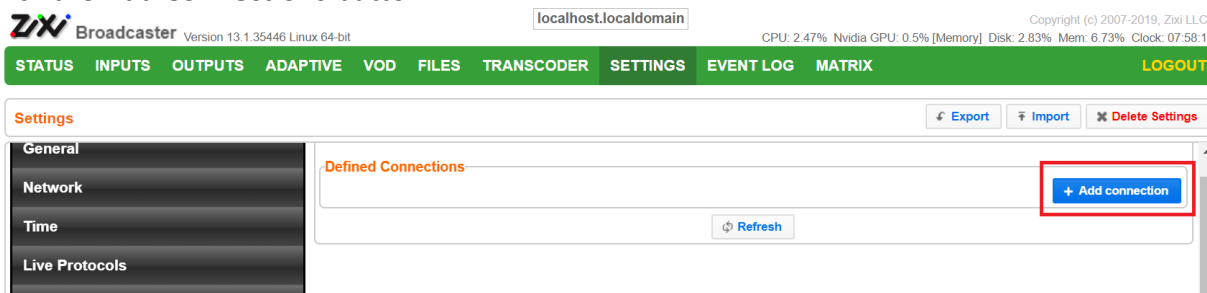
Status	Name	IP	SSH Key	Streams	CPU	RAM	Version	Actions
Ok	Demo	54.83.40.212	demo_ssh_key	59	13.41%	79.53%	1.13.1.36841	[Mute] [Edit] [Delete] [Refresh]

## SSH Connections

SSH (Secure Shell) is a network protocol that allows a secure access over an encrypted connection. Reverse SSH is a technique that can be used to access systems (that are behind a firewall) from the outside world. As part of the SSH configuration, you can configure a connection to the server and then configure any number of SSH tunnels to the same server.

### ➔ To create an SSH connection to a server:

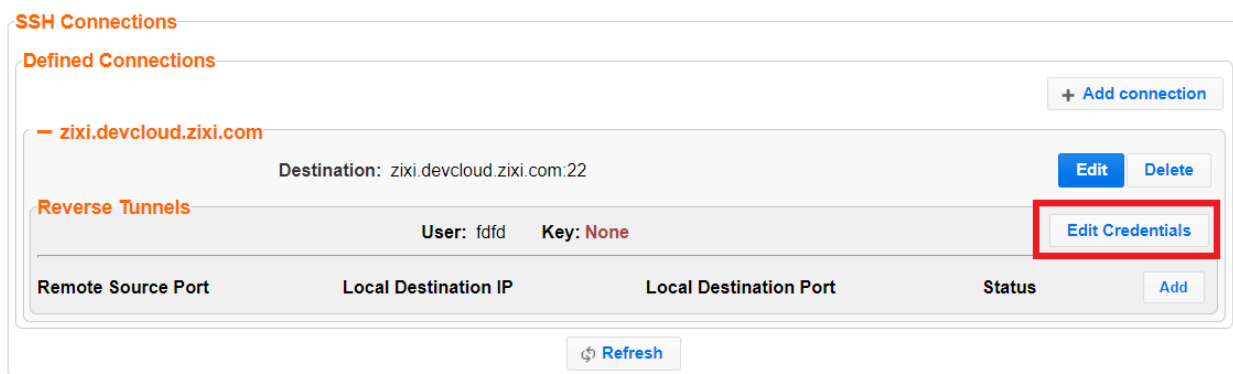
1. Click the **Add Connections** button.



2. In the **Host** field, enter the IP address or host name of the host that you wish to access.
  3. In the **Port** field, enter the port number to access the host.
  4. Click **Apply**.
- The connection to the specified server is configured.

### ➔ To create a reverse tunnel to the same server:

1. Under the destination that you have configured, click **Edit Credentials**.



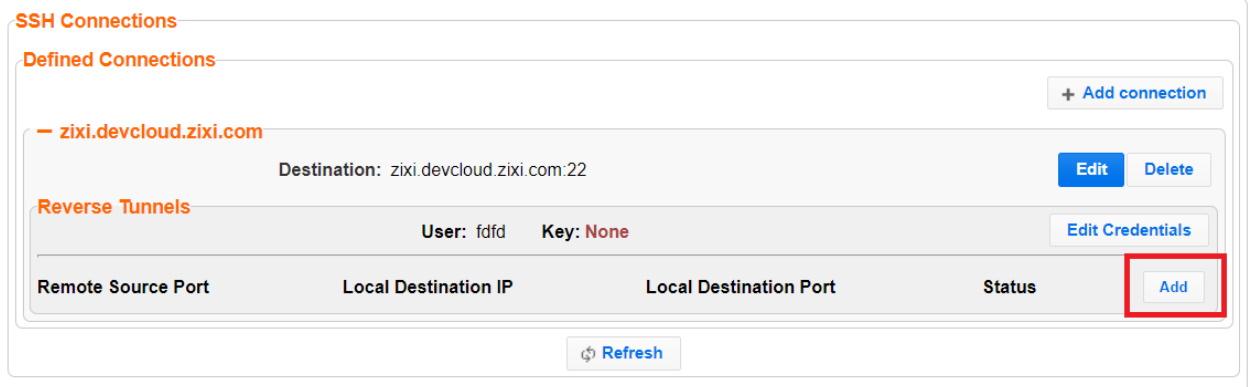
The **Connection Credentials** dialog appears.

2. In the **User** field, enter the user name to access the tunnel.
3. Click **Private Key**, navigate to the file location and select the file.
4. Click **Apply**.

## Server Settings

### SSH Connections

5. Click **Add** to add a reverse tunnel.



The screenshot shows the 'SSH Connections' interface. At the top, there is a section for 'Defined Connections' with a '+ Add connection' button. Below this, a connection is listed for 'zixi.devcloud.zixi.com' with a destination of 'zixi.devcloud.zixi.com:22' and buttons for 'Edit' and 'Delete'. Underneath is the 'Reverse Tunnels' section, showing 'User: fdfd' and 'Key: None' with an 'Edit Credentials' button. A table with columns 'Remote Source Port', 'Local Destination IP', 'Local Destination Port', and 'Status' is visible, with an 'Add' button highlighted in a red box at the bottom right of the table. A 'Refresh' button is located at the bottom center of the interface.

The **Tunnel details** dialog appears.

6. In the **Remote Source Port** field, enter the port number of the remote server.
7. In the **Local Destination IP** field, enter the IP address of the local server.
8. In the **Local Destination Port** field, enter the port number to connect the tunnel.
9. Click **OK**.
10. Repeat steps 5-9 to add additional tunnels.

# Authorization

Authorization can be configured for any incoming or outgoing streams.

Field	Description
<b>Authorization</b>	

**None (default)** - authorization will be defined in the Input and Output streams. In this case you will need to define the password in the Input and the Output streams. Unless you select the Allow automatic push inputs and/or Allow automatic pull outputs checkboxes (as described below).

**Text File** – this option manages authorizations in a text file. This option is useful for large OTT operations, as it does not require configuration for every user. The text file contains pairs of stream name and password that are used for the input streams. For the outputs, it contains the stream name, user and password of the viewer.

**Linux Users** – uses the built-in authorization of the Linux operating system.

**Global Password** – a single password that will be used for all access requests.

**Users Files** -

**HTTP Server** – Zixi Broadcaster will generate an HTTP request to an external server on every connection.

**Stream access** – access authorization on the stream level. The password needs to be the same as the password to access the stream.

**Windows Users** – uses the built-in authorization of the Windows operating system.

## Server Settings

---

### Authorization

---

---

Other override authorization techniques are available. For more information contact Zixi support

---

---

**Parameter** Specify the password that you would like to use for access

---

---

If this checkbox is selected, the Zixi Broadcaster will automatically create an input stream each time it receives a push stream from a source.

**Allow Automatic Push inputs** However, if this checkbox is not selected, only predefined Push Input streams with the defined credentials will be allowed in (this is the default setting and the behavior in previous versions).

Default: not selected.

---

---

**Allow Automatic pull outputs** If this checkbox is selected, any Zixi Receiver that requests a Pull output will be granted access. In this case, a Pull output will be automatically created in the Zixi Broadcaster (this is the default setting and the behavior in previous versions).  
However, if this checkbox is not selected, only predefined Pull output streams with the defined credentials will be pulled by the Zixi Receiver.

Default: selected.

---

---

**Separate user folders** Select this checkbox to enable access by user folders

---

---

**Logging** Select the checkbox to enable logging of the connections

---

---



# Logging

The Logging screen defines the logs location and preferences for debugging purposes.

Field	Description
<b>Logging</b>	
<b>General Level</b>	The Severity Level that will be collected in the log, from Info, Warning, Error, to Fatal. Selection of lower severity levels will include all higher-level logs. Default: None, since Logging is resource-intensive and only enabled when necessary.
<b>Path</b>	The log file name and path where the collected event information will be stored.
<b>Max log files</b>	The maximal number of log files.
<b>Max log file size (MB)</b>	The maximal size of each log file.
<b>Send crash reports</b>	Select the checkbox to send a crash report to Zixi Support.

## Server Settings

ASI

# ASI

The ASI screen is used to configure the ASI interface when there is an ASI card in the machine. You can use this screen to reverse the direction of the ports on the ASI card. The port numbers should correspond to the port numbers on your ASI card.

**ASI Settings**

**Available ports**

#	Name	Data direction	
1	Port 1	Input	→ <a href="#">Reset to output</a>
2	Port 2	Input	→ <a href="#">Reset to output</a>
3	Port 3	Output	← <a href="#">Reset to input</a>
4	Port 4	Output	← <a href="#">Reset to input</a>

ASI output stuffing:

ASI output PLL:

ASI fix time:

Initial buffer time[ms]:

Read buffer size[B]:

[Apply](#)

Field	Description
<b>ASI output stuffing</b>	Select this checkbox to achieve CBR (constant bitrate) by padding the ASI stream with TS null packets.
<b>ASI output PLL</b>	Select this checkbox to enable a phase-locked loop. A phase-locked loop or phase lock loop (PLL) is a control system that generates an output signal whose phase is related to the phase of an input signal. When selecting this option, the system will adjust the bitrate that is sent according to the fullness rate of the output buffer.
<b>ASI fix time</b>	Select this checkbox to adjust input timestamps to local machine time.
<b>Initial buffer time [ms]</b>	To avoid underflows, specify the output buffer size that will be filled before starting the transmission.
<b>Read buffer size [B]</b>	Specify the input read buffer to fill, before passing data to the application. Larger buffer size should be specified for higher bitrates; Smaller buffer size will be more responsive and smoother in lower bitrate streams.

ASI

### Reversing the directions of the ports

#### ➔ To Reverse the Direction of the Ports:

1. From the **Settings** page of the Zixi Broadcaster, click **ASI** on the left-hand side.

The various **ASI settings** are displayed.

Click the desired **RESET** button to the right of the port that you want to reverse.

A confirmation window is displayed with the notification that the service must first be restarted before the direction is reversed.

Field	Description
<b>Available ports</b>	
<b>Name</b>	Displays the port number used by the ASI protocol.
<b>Data Direction</b>	Displays the direction of the data being transmitted by the ASI protocol.

# Threads

The Threads screen enables the configuration of the maximal number of concurrent CPU threads allowed for each component/service. Enter -1 to limit the threads according to the number of cores.

The screenshot shows the Zix Broadcaster web interface. At the top, there is a navigation bar with the following menu items: STATUS, INPUTS, OUTPUTS, ADAPTIVE, VOD, FILES, TRANSCODER, SETTINGS, EVENT LOG, MATRIX, and LOGOUT. The 'SETTINGS' menu item is highlighted. Below the navigation bar, there is a 'Settings' section with buttons for 'Export', 'Import', and 'Delete Settings'. On the left side, there is a sidebar menu with the following items: General, Network, Time, SNMP, Live Protocols, Multicast Pool, File Transfer & VOD, Cluster, ZEN Master, SSH Connections, Authorization, Logging, ASI, Threads, and SCTE35 Report. The 'Threads' item is selected. The main content area is titled 'Threads settings' and contains the following configuration options:

Input:	<input type="text" value="-1"/>
Output:	<input type="text" value="-1"/>
VOD:	<input type="text" value="4"/>
Read:	<input type="text" value="1"/>
Pool:	<input type="text" value="2"/>
RTMP:	<input type="text" value="-1"/>
UDP In:	<input type="text" value="-1"/>
UDP Out:	<input type="text" value="-1"/>
HTTP In:	<input type="text" value="1"/>
Transcoder:	<input type="text" value="4"/>
Push Output Receiver:	<input type="text" value="2"/>
HTTP:	<input type="text" value="-1"/>
Authorization:	<input type="text" value="1"/>
Analyzer:	<input type="text" value="2"/>

-1 means number of cores.

The following elements can be configured:

- Input
- Output
- VOD
- Read
- Pool
- RTMP
- UDP In
- UDP Out
- HTTP In
- Transcoder
- Push Output Receive
- HTTP
- Authorization
- Analyzer

## Server Settings

### SNMP (Linux Only)

## SNMP (Linux Only)

The SNMP screen includes basic configuration parameters for the configuration of Simple Network Management Protocol, which enables monitoring and control of Zixi Broadcaster by a monitoring system. For further information on how to enable and configure SNMP on Zixi Broadcaster, see Configuring SNMP section.

Settings

Export Import Delete Settings

General  
Network  
Time  
SNMP  
Live Protocols  
Multicast Pool  
File Transfer & VOD  
Cluster  
SSH Connections  
Authorization  
Logging  
ASI

Service: On

Read only community name: .....

Read-write community name: .....

SNMPv3 Users

User	Write access
------	--------------

Trap receivers

Host	Port
10.7.0.23	162

Apply

Field	Description
<b>Service</b>	Click the button to Start/Stop the SNMP service.
<b>Read-only Community Name</b>	Used in SNMPv1 and SNMPv2, this parameter is used to specify the Community Name to enable read-only access to the Zixi Broadcaster monitored information. If the community string is correct, Zixi Broadcaster will respond with the requested information. If the community string is incorrect, Zixi Broadcaster will simply ignore the request and will not respond.
<b>Read-write Community Name</b>	Used in SNMPv1 and SNMPv2, this parameter is used to specify the Community Name to enable read-write access to Zixi Broadcaster. Read-write objects include, for example, zbiActive, which indicates and/or sets the Active state of an Input stream.
<b>SNMP V3 Users</b>	<p>For SNMPv3 monitoring systems, click the + button to add authentication credentials and fill-in the following fields:</p> <ul style="list-style-type: none"><li>User name - enter the SNMP user name</li><li>Write access - select the checkbox to allow Read-Write access rights. If the checkbox is not selected, the user will have Read-only access rights.</li></ul>

## Server Settings

---

### SNMP (Linux Only)

---

---

- Password - enter the SNMP user's password.
- 
- 

Specify SNMP Trap receivers that will listen to the SNMP Traps that will be sent by Zixi Broadcaster. Click the + button to add authentication credentials of the Trap Receiver, by filling-in the following fields:

- Trap Receivers**
- Community - Specify the community name for access.
  - Host - Specify the IP address to which the Trap messages will be transmitted.
  - Port - Specify the Port number to which the Trap messages will be sent transmitted.
- 
-

# Viewing Events Log

The Event Log screen displays all the events (errors, connections, configuration changes, analysis events etc.) that occurred in the Zixi broadcaster. The logs are displayed in the server time (not local time) and can be exported as a .cvs file.

In Linux, the event log is written to a file located in the following location on the disk:  
`/log/var/messages`

### ➔ To Clear the Log:

In the **Event Log** screen, click **Clear** on the upper right-hand corner. The events on the log screen are deleted.

### ➔ To Save the Log as a File:

In the **Event Log** screen, click **Save** on the upper right-hand corner. A .cvs file is created and saved to the local destination.

### ➔ To Refresh the Screen:

In the **Event Log** screen, click **Refresh** on the upper right-hand corner. The data on the screen is updated.

# Using the Matrix

The Zixi Matrix view provides a visual-based alternative to managing the streams, by previewing the input and output streams with their current status. Matrix offers drag and drop capabilities to connect input streams to outputs as well as some of the actions that were covered in the Performing Actions on Input Streams and Performing Actions on Output Streams sections above.

## Connecting Input to Output

An easier method of connecting inputs to outputs.

### ➔ To Connect an Input to an Output:

Drag the desired input thumbnail and drop it over a specific output thumbnail. The desired input is connected to the selected output.



## Disconnecting Output from Input

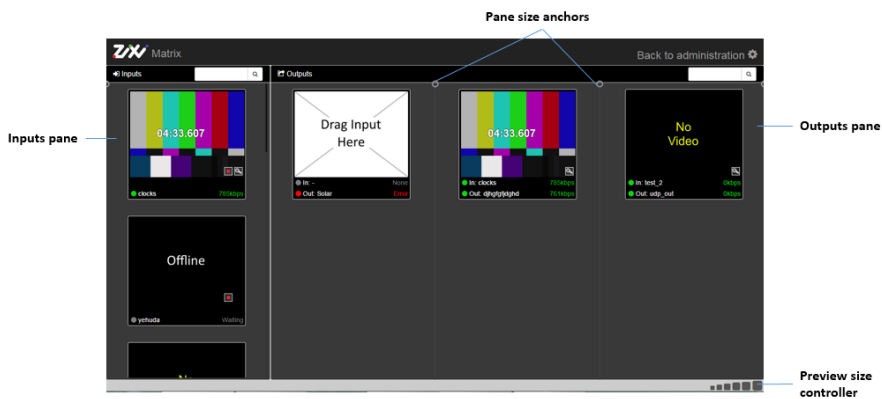
### ➔ Disconnect an Input to an Output:

1. Click on the **To connected stream** that you want to disconnect.  
The **Input stream data** window is displayed.
2. Click **Unlink**. A confirmation window is displayed.
3. Click **Confirm**. The Output is disconnected from the Input.

## Matrix Interface Elements

The Matrix view displays preview thumbnails of the Input and Output streams. Preview thumbnails of input streams are presented in the Inputs pane and preview thumbnails of output streams are presented in the Outputs pane.

- If the stream is active - the live video will be shown in the preview thumbnail.
- If an input stream is not active – the preview thumbnail will include an “Offline” message.
- If an output stream has been created, but has not been connected to an input – the preview thumbnail will include the following message - “Drag Input Here”.
- If a connected output stream does not receive a live stream - the preview thumbnail will include a “No Video” message.



## Changing Matrix View

You can modify the size of the panes, the size of the preview thumbnails, and re-arrange the order the preview thumbnails.

### ➔ To modify the size of the panes:

Click on one of the rounded **pane size anchors** (see screenshot above)

### ➔ To modify the size of the preview thumbnails:


Click on one of the sizes in the **preview size controller** (see screenshot above).

### ➔ To re-arrange the order the preview thumbnails:

Drag and drop the preview thumbnail to the desired position within the pane.

## Searching for Specific Streams

### ➔ To Search for a Specific Stream:

1. Type in the specific stream name in the appropriate (Input or Output) search text box on the top of the console and click .
2. The relevant stream will be the only one to be displayed in the pane.

## Preview Thumbnail Elements

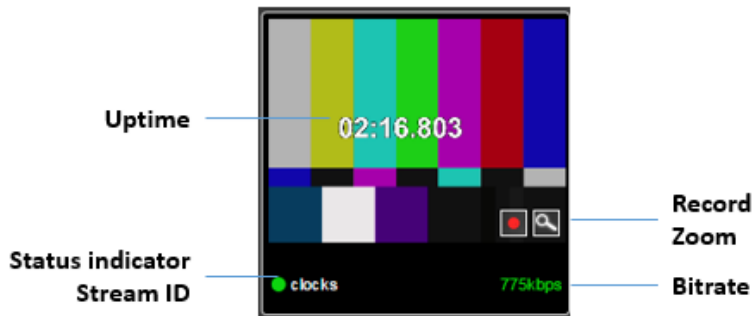
The following stream formats are supported for thumbnail preview:

- Scan Style: Interlaced and Progressive
- Codec: MPEG2, H.264/AVC
- Frame Size: SD, HD (720p), Full HD (1080p), 4K (2160p)
- Frame Rate: 24fps, 25fps, 29.97fps, 30fps, 59.94fps, 60fps
- Chroma Subsampling: 4:2:0
- Bit Depth: 8-bit (MPEG2, H.264/AVC), 10-bit (H.264/AVC)

## Using the Matrix

### Viewing Stream Details

The preview thumbnail includes the following elements:



Element	Description	Action
<b>Status Indicator</b>	Displays the unique identifier for this group.	
<b>Stream ID</b>	Displays the unique name for the stream.	
<b>Bitrate [kbps]</b>	Displays the bitrate of the active stream.	
<b>Uptime</b>	Displays the amount of time the stream has been active.	
<b>Record</b>	Stream recording	Click the record button to start recording. Click again to stop recording
<b>Zoom</b>	Opens the stream in a larger preview window with sound	Click the Zoom icon to open the larger preview screen.

## Viewing Stream Details

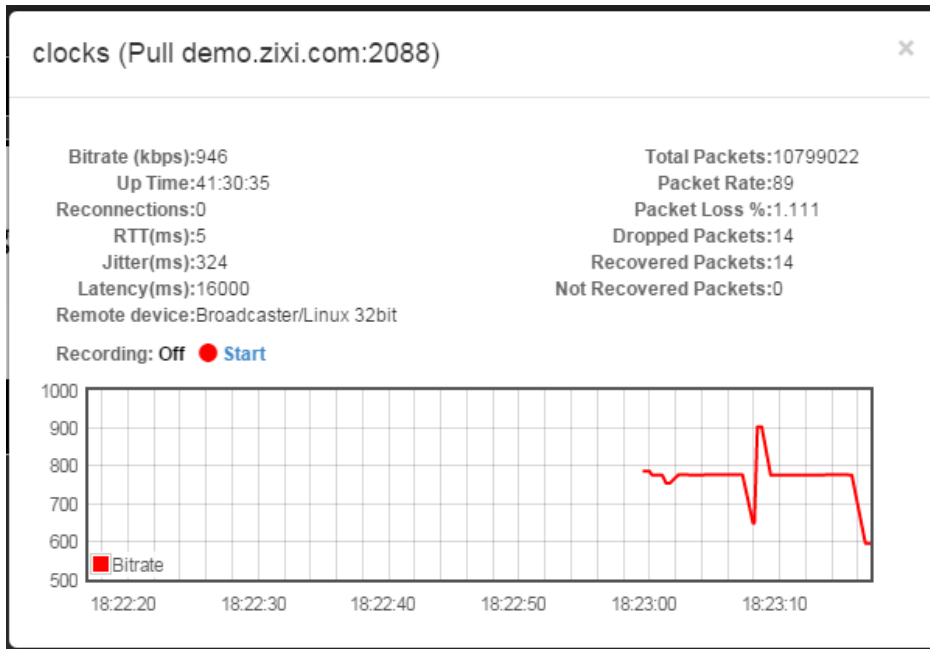
### ➔ To View Stream Details:

Click anywhere on the preview thumbnail.

The Stream Details window opens:

## Using the Matrix

### Viewing Stream Details



# Configuring SNMP

Zixi Broadcaster enables monitoring and control via SNMP (on Linux only). The monitoring system can receive SNMP Trap messages from Zixi Broadcaster as well as access monitored objects with read-only and read-write access privileges (objects and their access permissions are defined in the MIB). Zixi Broadcaster supports SNMPv1, SNMPv2, and SNMPv3. Before using SNMP you need to ensure that SNMP is enabled in your Zixi Broadcaster license, run a script, as described below, and configure the SNMP settings in the Settings >SNMP screen.



The SNMP feature is not enabled by default. If you do not see this screen, please contact Zixi Support.

## Activating SNMP

### ➔ To Activate SNMP:

Run the script file `./install_snmp.sh`, located in the Broadcaster folder, where Zixi Broadcaster was installed. The events on the log screen are deleted.



Note: Running the script may cause Zixi Broadcaster to restart.

## Configuring SNMP Settings

### ➔ To configure SNMP:

1. In Zixi Broadcaster UI, go to **Settings > SNMP**.

The screenshot shows the 'Settings > SNMP' configuration interface. On the left is a navigation menu with options: General, Network, Time, SNMP (selected), Live Protocols, Multicast Pool, File Transfer & VOD, Cluster, SSH Connections, Authorization, Logging, and ASI. The main content area has a top bar with 'Export', 'Import', and 'Delete Settings' buttons. Below this, the 'Service' is set to 'On'. There are two fields for 'Read only community name' and 'Read-write community name', both containing asterisks. Under 'SNMPv3 Users', there is a table with columns 'User' and 'Write access', and a '+' button. Under 'Trap receivers', there is a table with columns 'Host' and 'Port', containing the entry '10.7.0.23' and '162', with edit and delete icons. An 'Apply' button is at the bottom.

User	Write access
------	--------------

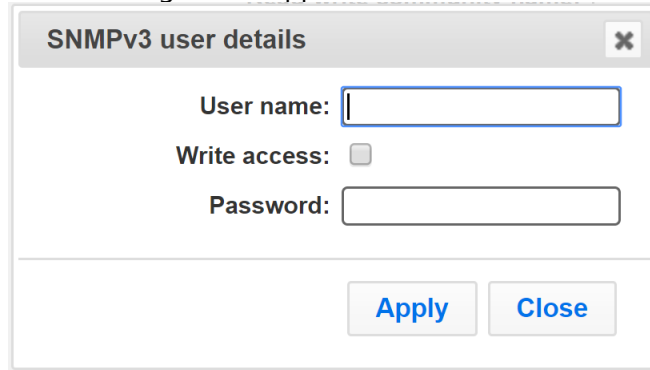
Host	Port
10.7.0.23	162

## Configuring SNMP

### Configuring SNMP Settings

2. In the **Service** field, click the button to activate the service.
3. Do one of the following:
  - For SNMPv1 or SNMPv2, fill in the read-only and/or read-write community name in the **Read only community name**, and **Read-write community name** fields.
  - For **SNMPv3**, under the SNMPv3 Users section, click the + button.

The following screen appears:

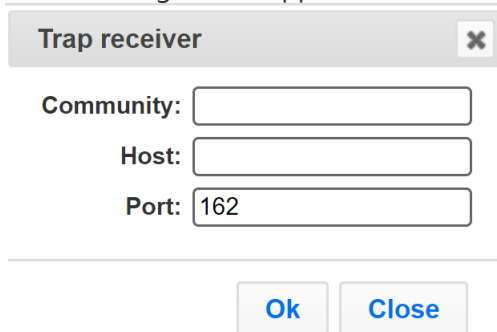


The dialog box titled "SNMPv3 user details" contains the following fields and controls:

- User name:** A text input field.
- Write access:** A checkbox.
- Password:** A text input field.
- Buttons:** "Apply" and "Close" buttons.

- In the **User Name** field, enter the SNMP user name.
  - In the **Write Access** field, select the checkbox to allow **Read-Write** access rights. If the checkbox is not selected, the user will have **Read-only** access rights.
  - In the **Password** field, enter the SNMP user's password.
4. Under the **Trap Receivers** section, specify SNMP Trap receivers that will listen to the SNMP Traps that will be sent by Zixi Broadcaster, by clicking the + button to add a trap receiver.

The following screen appears:



The dialog box titled "Trap receiver" contains the following fields and controls:

- Community:** A text input field.
- Host:** A text input field.
- Port:** A text input field with the value "162" pre-filled.
- Buttons:** "Ok" and "Close" buttons.

5. In the **Community** field, specify the community name for access.
6. In the **Host** field, specify the IP address to which the Trap messages will be transmitted.
7. In the **Port** field, specify the Port number to which the Trap messages will be sent transmitted.
8. Click **Apply**.

## Viewing MIB Information

You can view the "Root MIB" (which defines the Zixi Broadcaster tree root definitions) and the "Broadcaster MIB" (which describes the Zixi Broadcaster MIB tree definition, including all its objects).

### ➔ To view MIB information:

1. In Zixi Broadcaster UI, go to **Settings > SNMP**.
2. At the upper right-hand corner click the **Root MIB** and/or **Broadcaster MIB** links.

Settings

Export Import Delete Settings

General  
Network  
Time  
SNMP  
Live Protocols  
Multicast Pool  
File Transfer & VOD  
Cluster  
SSH Connections  
Authorization  
Logging  
ASI

Service: On

Read only community name: .....

Read-write community name: .....

SNMPv3 Users

User	Write access	
		+

Trap receivers

Host	Port	
10.7.0.23	162	+ ✎ ✕

Apply

Root MIB  
Broadcaster MIB

## SNMP Alerts/Notifications

The SNMP traps are sent when an alert is raised and when alert is cleared on following occasions:

- Input stream alerts ( zbtInputStreamAlertRaised = .1.3.6.1.4.1.38369.1.1.1.0.58; zbtInputStreamAlertCleared = .1.3.6.1.4.1.38369.1.1.1.0.60)  
Important parameters:  
\$ zbsProgramID ( Id of the broadcaster that sends the trap, .1.3.6.1.4.1.38369.1.1.1.1.2)  
\$ zbtStreamID ( Id of the stream, .1.3.6.1.4.1.38369.1.1.1.0.2)  
\$ zbtStreamAlertType ( alert type, .1.3.6.1.4.1.38369.1.1.1.0.10)
- TR101 analyser alerts ( zbtTR101AlertRaised = .1.3.6.1.4.1.38369.1.1.1.0.90; zbtTR101AlertCleared = .1.3.6.1.4.1.38369.1.1.1.0.91)  
Important parameters:

## Configuring SNMP

---

### SNMP Alerts/Notifications

- \$ zbsProgramID ( Id of the broadcaster that sends the trap, .1.3.6.1.4.1.38369.1.1.1.1.2)
- \$ zbtStreamID ( Id of the stream, .1.3.6.1.4.1.38369.1.1.1.0.2)
- \$ zbtTR101AlertErrorCode ( TR101 alert code, .1.3.6.1.4.1.38369.1.1.1.0.30)
- Output stream alerts ( zbtOutputStreamAlertRaised  
= .1.3.6.1.4.1.38369.1.1.1.0.59; zbtOutputStreamAlertCleared  
= .1.3.6.1.4.1.38369.1.1.1.0.61)
- \$ zbsProgramID ( Id of the broadcaster that sends the trap, .1.3.6.1.4.1.38369.1.1.1.1.2)
- \$ zbtStreamID ( Id of the input stream where output is  
assigned, .1.3.6.1.4.1.38369.1.1.1.0.2)
- \$ zbtOutputID ( Id of the output, .1.3.6.1.4.1.38369.1.1.1.0.3)
- \$ zbtStreamAlertType ( alert type, .1.3.6.1.4.1.38369.1.1.1.0.10)
- Performance alerts ( zbtPerformanceAlertRaised  
= .1.3.6.1.4.1.38369.1.1.1.0.80; zbtPerformanceAlertCleared  
= .1.3.6.1.4.1.38369.1.1.1.0.81)

Important parameters:

- \$ zbsProgramID ( Id of the broadcaster that sends the trap, .1.3.6.1.4.1.38369.1.1.1.1.2)
- \$ zbtPerformanceAlertType (alert type, .1.3.6.1.4.1.38369.1.1.1.0.20).

In addition, traps are sent on following events:

- Input stream created ( zbtInputStreamCreated = .1.3.6.1.4.1.38369.1.1.1.0.54)
- Output stream created ( zbtOutputStreamCreated = .1.3.6.1.4.1.38369.1.1.1.0.55)
- Input stream up ( zbtInputStreamUp = .1.3.6.1.4.1.38369.1.1.1.0.50)
- Input stream down ( zbtInputStreamDown = .1.3.6.1.4.1.38369.1.1.1.0.50)
- Output stream up ( zbtOutputStreamUp = .1.3.6.1.4.1.38369.1.1.1.0.51)
- Output stream down ( zbtOutputStreamDown = .1.3.6.1.4.1.38369.1.1.1.0.53)
- Input stream deleted ( zbtInputStreamDeleted = .1.3.6.1.4.1.38369.1.1.1.0.56)
- Output stream deleted ( zbtOutputStreamDeleted = .1.3.6.1.4.1.38369.1.1.1.0.57)
- Successful broadcaster process start ( zbtProgramStartSuccess  
= .1.3.6.1.4.1.38369.1.1.1.0.200)
- Failure to start broadcaster process ( zbtProgramStartFailed  
= .1.3.6.1.4.1.38369.1.1.1.0.201)

Important parameters:

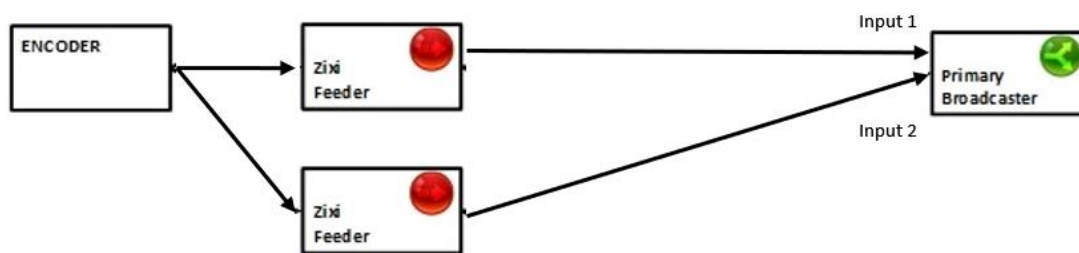
- \$ zbsProgramID ( Id of the broadcaster that sends the trap, .1.3.6.1.4.1.38369.1.1.1.1.2)
- \$ zbtStreamID ( Id of the input stream, .1.3.6.1.4.1.38369.1.1.1.0.2) applicable for inputs' and outputs' events.
- \$ zbtOutputID ( Id of the output, .1.3.6.1.4.1.38369.1.1.1.0.3), applicable only for outputs' events.



# Configuring Hitless Failover

Zixi features hitless failover for inputs, enabling uninterrupted streaming when switching from one source to another. The hitless failover feature is based on the SMPTE 2022-7 standard, which specifies “seamless” or hitless failover between binary-identical streams with synchronized RTP headers. Zixi's patent-pending hitless failover technology works by comparing the transport stream payload, without relying on sequences being synchronized. Zixi then applies a proprietary sequencing algorithm to identify missing parts and fill-in missing packets.

The Hitless Failover feature is supported in the inputs of Zixi Feeder, Broadcaster, and Receiver.



The hitless failover configuration involves grouping two inputs in a "Failover Group". The failover group includes the following configuration options for the merging of binary-identical streams: The merging of the streams can be achieved using one of the following methods:

- **RTP headers (SMPTE 2022-7)** – if the sources are RTP and the RTP headers were maintained, Zixi will use the SMPTE 2022-7 to compare and merge the two streams in order to allow hitless failover. This option is the most recommended as it will use the sequence and the timestamp in the packets to compare which packets are missing. This method is more accurate than methods that compare the entire payload or methods that use sophisticated DNA sequencing algorithm, which are more error prone, as the payload may have multiple matches.
- **RTP headers or TS content** – if the sources do not have RTP headers, Zixi will try to merge using the MPEG-TS content itself.

In these cases, when one of the streams is disrupted or interrupted completely, Zixi Broadcaster will seamlessly use the packets of the other input without any disruptions.



If the streams are not identical, the failover process will not be seamless and may take up to a second.

A Failover Group can comprise two or more inputs. In a group of three or more, you can define one or more streams as a "Backup". In this case the system will always try to recover packets from "Primary" streams first and only if there are no available primary streams, it will use the backup

## Configuring Hitless Failover

---

### SNMP Alerts/Notifications

streams. When one of the primary streams becomes available, the system will automatically jump to the primary stream.

While the Broadcaster GUI only enables configuration of primary streams at a single priority level, through the Zixi Broadcaster API it is possible to assign different priorities as follows:

- Priority 0 – means backup. It will be used only if all other streams are unavailable.
- Priority greater than 0 - means primary.

The Failover Group has its own Stream ID, just like an Input Stream, and will appear in the Inputs list. Although the topology above shows only one Zixi Broadcaster, it is also possible to have multiple targets (e.g. two or more Zixi Broadcasters), each with its own Failover Group of inputs to enable a more complete system-level redundancy.

To configure a Failover Group, follow the instructions in the [Creating a Failover Group](#) section.

# Configuring HTTPS

Zixi Broadcaster supports HTTPS encryption based on OpenSSL. Zixi Broadcaster expects certificate files in X.509 textual format (sometimes called 'pem') with a full chain of certificates, each one starts with "-----BEGIN CERTIFICATE-----", and the associated private key file, that starts with "-----BEGIN PRIVATE KEY-----". The certificate can be provided by an SSL/TLS certificate provider (Certification Authority) where the domain is registered, such as VeriSign, Digicert, etc.. Alternatively, the certificate can be self-signed. A self-signed certificate is a certificate that is signed with its own private key. Self-signed certificates can be used to encrypt data just as well as CA-signed certificates, but your users will be displayed a warning that says that the certificate is not trusted by their computer or browser. Therefore, self-signed certificates should only be used if you do not need to prove your service's identity to its users (e.g. non-production or non-public servers).

## Obtaining a CA-signed Certificate

Reach out to the hosting provider for instructions on how to obtain a CA-signed HTTPS certificate. As part of this process, you will be required to generate a certificate signing request (CSR). A CSR consists mainly of the public key of a key pair, and some additional information. Both of these components are inserted into the certificate when it is signed. To learn more about generating a CSR, go to - <https://www.digitalocean.com/community/tutorials/openssl-essentials-working-with-ssl-certificates-private-keys-and-csrs>.

## Creating a Self-signed Certificate

Instead of obtaining a CA-signed certificate, you can create a self-signed certificate.

You can generate self-signed x.509 certs with OpenSSL by using the following command:

```
openssl req -x509 -newkey rsa:2048 -keyout selfsigned_key.pem -out selfsigned_cert.pem
```

To learn more about generating a self-signed certificate, go to:

<https://www.digitalocean.com/community/tutorials/openssl-essentials-working-with-ssl-certificates-private-keys-and-csrs>.

## Uploading the Certificate and Private Key

After obtaining a CA-signed certificate or creating a self-signed one, you will need to upload the certificate and private key to Zixi Broadcaster.

## Configuring HTTPS

---

### Uploading the Certificate and Private Key

➔ **To upload the certificate and private key:**

1. In **Zixi Broadcaster UI**, go to **Settings > General**.
2. Click + to expand the HTTPS and DTLS Streaming Certificate section
3. In the **Certificate Uploaded** field, click **Upload** and select the certificate file.
4. In the **Private Key Uploaded** field, click **Upload** and select the Private Key.
5. In the **Private Key Passphrase** field, type the passphrase or leave empty (Default passphrase is empty, ignore the '\*\*\*\*').
6. Click **Apply**.  
If the process was successful, the **SSL setup status** field will turn to "OK".

# Configuring NDI Streams

Network Device Interface (NDI) is a royalty-free software standard developed by NewTek to enable video-compatible products to communicate, deliver, and receive broadcast-quality video in a high-quality, low-latency manner that is frame-accurate and suitable for switching in a live production environment.

NDI was designed to deliver 1080i HD video at VBR data rates typically around 100 Mbit/s. Zixi Broadcaster uses an NDI library to read uncompressed frames from the NDI stream and pushes them to the Zixi Broadcaster's transcoder. The transcoder can then transcode the uncompressed stream into a transport stream using one of the available codecs, such as H.264 or H.265 video and AAC or Opus audio.

Zixi Broadcaster supports the following flows for NDI:

- **NDI Input** – the NDI source stream, which is uncompressed, is transcoded via the Zixi Broadcaster transcoder into any one of the available codecs. The transcoded stream will be available as an Input Stream when creating an Output in Zixi Broadcaster.
- **NDI Output** – Zixi Broadcaster can transcode a TS stream into an NDI output using Raw video and audio profiles. The NDI output is configured in the transcoding profile itself, so there is no need to create an NDI Output. When the input stream is transcoded, it becomes available to be pulled by NDI enabled devices.

A typical deployment involves ingesting NDI to a Zixi Broadcaster through LAN at one location and then using Zixi Broadcaster to compress the stream into high quality AVC/H.264 or HEVC/H.265. The stream is then delivered to another location over the public internet to any distance using the reliable, low latency Zixi protocol and then at the receiving location the stream is decoded by a Zixi Broadcaster and broadcasted on the local LAN as NDI again. While a single 1920×1080@30 fps NDI stream may require 125 Mbps of bandwidth on the local network, once it is compressed for delivery with Zixi it will only require 10 to 15Mbps using HEVC/H.265 and no perceptible quality loss, to move the stream over the public internet.

## NDI Stream Discovery

NDI uses the mDNS (Bonjour / Zeroconf) discovery mechanism to create a zero-configuration environment for discovery of NDI streams available on the LAN. Through mDNS, devices such as the Zixi Broadcaster can discover available NDI streams for processing. The mDNS (Bonjour / Zeroconf) discovery mechanism should be installed as required according to the system's OS, as described in the OS Support section. This means that there are different requirements depending on the type deployment:

- **Using NDI within a local area network** – on a local area network NDI will use the discovery mechanism to advertise the sources and Zixi Broadcaster will automatically

## Configuring NDI Streams

---

### NDI Stream Discovery

discover these streams, while listing them in the **Stream** parameter of the NDI input stream.

- **Using NDI in the cloud** – to make the streams discoverable in the cloud, you need to deploy an NDI Discovery Server and configure the NDI plug-in with the settings included in the following example:

```
{
  "ndi": {
    "groups": {
      "recv": "\"public\"",
      "send": "\"public\""
    },
    "multicast": {
      "recv": {
        "enable": false,
        "subnets": ""
      },
      "send": {
        "enable": false,
        "netmask": "255.255.255.0",
        "netprefix": "239.192.136.100"
      }
    },
    "networks": {
      "discovery": "22.127.94.57",
      "ips": ""
    },
    "tcp": {
      "recv": {
        "enable": false
      },
      "send": {
        "enable": false
      }
    },
    "unicast": {
      "recv": {
        "enable": false
      },
      "send": {
        "enable": false
      }
    }
  }
}
```

## OS Support

The use of mDNS requires operating system support as follows:

- **Windows** - Windows 10 has mDNS support built in, which is necessary for any NDI applications running on Windows.
- **CentOS 7** - The recommended platform for the Zixi Broadcaster is CentOS 7. mDNS support is provided by the Avahi package which typically is not installed by default on CentOS 7.

➔ **To install Avahi, run the following commands on the server hosting the Zixi Broadcaster:**

```
# sudo yum install avahi
# sudo service avahi-daemon start
```

## Enabling Multicast Network Support

The mDNS Ethernet frame is a multicast UDP packet that broadcasts to:

- MAC address 01:00:5E:00:00:FB(for IPv4)
- IPv4 address 224.0.0.251
- UDP port 5353

You need to verify that the LAN where NDI streams are present is multicast enabled and that all systems needing access to NDI streams, such as the Zixi Broadcaster, do not block mDNS via their firewall.

## NDI config file location

- On Windows: %PROGRAMDATA%\NewTek\NDI\ndi-config.v1.json
- On Linux: \$HOME/.newtek/ndi-config.v1.json



On Linux the NDI config files should be placed in the home directory of the user which Broadcaster is running as root. Thus, the file must be located at /root/.newtek/ndi-config.v1.json.

## Specifying the NDI Discovery Server

➔ **To specify NDI Discovery Server:**

1. Open ndi-config.v1.json.
2. In the "networks" section, under "discovery" add the discovery server's IP, which is the address of a host where NDI Discovery Server is working. For example, if you have it working on 192.168.0.253, then your config will have this section:

## Configuring NDI Streams

### Configuring Transcoding Profiles for NDI Sources

```
"networks": {"ips": "", "discovery": "192.168.0.253"},
```

## Configuring Transcoding Profiles for NDI Sources

Before creating the NDI Input in Zixi Broadcaster, you must configure at least one transcoding profile that meets the following requirements:

- **Disable Copy GOP from source** - NDI sources can be transcoded using any of the available transcoding profiles, however it is **NOT possible** to enable the **Copy GOP from source** feature. Since all frames in the original NDI source are uncompressed, Zixi cannot know which frames were i-frames in the stream, so the GOP length should be configured in the **GOP [frames]** parameter of the transcoding profile.
- **No Keep Original** - when configuring the stream, it is possible to remove the video or audio from the stream, however it is **NOT possible** to keep the original video or audio, as they are originally uncompressed.
- **No RAW Video/Audio** - transcoding profiles with **RAW Video** and/or **RAW Audio** will be filtered out and will not appear in the **Video profile/Audio Profile** fields on the NDI Input screen.

For detailed instructions on how to configure transcoding profiles, see [Transcoder](#) section.

## Adding an NDI Input to Zixi Broadcaster

The NDI is pulled into Zixi Broadcaster using the auto-discovery mechanism described in the NDI Stream Discovery section.

### ➔ To Add an NDI Input Stream:



Before adding the NDI Input Stream, you must configure the relevant transcoding profiles – See [Configuring Transcoding Profiles for NDI Sources](#).

1. In the Zixi Broadcaster Navigation menu, click the **Inputs** tab.
2. In the Menu bar, click **New Input**. The **Add a new input stream** window is displayed.



## Configuring NDI Streams

### Adding an NDI Input to Zixi Broadcaster

3. Select **NDI** for the input stream.

The screenshot shows the 'Add new input stream' dialog box. At the top, there are radio buttons for 'Push', 'Pull', 'UDP', 'Delayed', 'File', 'RTMP', 'RTSP', 'HLS', 'TS over HTTP', 'SRT', 'SRT Client', 'RIST', and 'NDI'. The 'NDI' radio button is selected. Below this, there are sections for 'Stream parameters', 'Enable time shift', 'Transmit as multicast', 'Enable encryption', and 'Recording parameters'. Each section has a checkbox. The 'Stream parameters' section includes a 'Stream' dropdown menu (set to 'KOBYS-IPHONE (N)'), 'Remove stream' dropdowns, and input fields for 'Audio Reference Level [dB]', 'Transport stream bitrate [kbps]', and 'Smoothing buffer [ms]'. There are also checkboxes for 'Use Intel hardware acceleration (if available)' and 'Smoothing buffer [ms]'. At the bottom right, there are 'Ok' and 'Cancel' buttons.

4. Select the maximum concurrent connections to this stream from the **Max. Outputs** drop-down list. Default: Unlimited.
5. Select the **Show in Matrix** check box to display this stream in the Matrix (see [Using the Matrix](#)). Default: Selected.
6. Specify the **Stream Parameters** (see Input Stream Parameter table).
7. If you want to **Enable time shift**, select this checkbox (see Input Stream Parameter table). The stream will be recorded and broadcasted after the specified delay. The time shift feature is typically used to broadcast a live event across time zones (due to the time differences between locations).
8. If you want to **Transmit as multicast**, select this checkbox (see Input Stream Parameter table).
9. If you want to configure the **Recording Parameters**, specify them in the respective fields (see Input Stream Parameter table).
10. If you want to **Enable encryption**, select this checkbox (see Input Stream Parameters Table). For more information, see [Encrypting an Input Stream in Zixi Broadcaster](#).
11. Click **OK**. The “Input added” message appears on the top of the screen. The NDI Stream is added to the Zixi Broadcaster and now appears in the list of streams available in Zixi Broadcaster Inputs screen.

#### Input Stream Parameters table


Parameter	Description
<b>Stream Parameters</b>	

## Configuring NDI Streams

---

### Adding an NDI Input to Zixi Broadcaster

---

<b>Stream</b>	Select one of the NDI streams available based on your configuration from the drop-down menu. On a local area network NDI will use the discovery mechanism to advertise the sources and Zixi Broadcaster will automatically discover these streams. To make the streams discoverable in the cloud, you need to deploy an NDI Discovery Server and configure the NDI plug-in with the settings mentioned above.
<b>Video Profile</b>	Select an existing transcoding video profile, or select <b>Remove stream</b> if you want to remove the video track.
<b>Audio Profile</b>	Select an existing transcoding audio profile, or select <b>Remove stream</b> if you want to remove the audio track.
<b>Audio Reference Level [dB]</b>	This parameter attenuates the audio level by the specified level in dB. This setting is used to prevent audio clipping.
<b>Transport Stream Bitrate [kbps]</b>	Specify the bitrate of the transport stream. This parameter is optional. If you do not specify a TS Bitrate, the stream will be VBR. If you specify a bitrate, the stream will be CBR. The bitrate value must be compatible with the selected profile.
<b>Use Intel hardware acceleration (if available)</b>	Select this checkbox to use the Intel hardware acceleration component if it available.
<b>Smoothing buffer</b>	The Smoothing feature uses a buffer to time packets according to the stream's clock. This is useful when there are bursts in the rate in which the stream arrives. The smoothing features creates a queue in the buffer, allowing smoother entry into the Broadcaster. The Smoothing parameter defines the size of the buffer in Milliseconds. A larger the buffer enables smoother entry. However, it may influence the latency rate.
<b>Enable time shift</b>	Select this checkbox to record the stream content and broadcast it after a specified delay. To deploy an input stream that uses the current stream's time shift, you need to configure a "Delayed" input stream. (See <a href="#">Adding Delayed Streams</a> ).
<b>Maximum delay [hh:mm:ss] (for Time Shift enabled)</b>	Specify the time of a delayed transmission of this stream. Maximum: 24 hours
<b>Transmit as multicast</b>	Select this checkbox to enable the transmission of this stream's outputs as multicast. A Zixi Broadcaster that will pull the stream will receive it in multicast. By default Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling <b>Multicast Pull</b> in the <b>Settings</b> .
	<div style="border: 1px dashed gray; padding: 5px;"> This setting can be modified without interrupting an existing stream.</div>

---

## Configuring NDI Streams

---

### Adding an NDI Input to Zixi Broadcaster

---

---

**Multicast-Only (for multicast enabled)** If desired, specify multicast-only outputs.

---

**Enable Encryption** Select this checkbox to encrypt the Input stream. For more information, see [Encrypting an Input Stream in Zixi Broadcaster](#).

---

**Encryption type (for Encryption enabled)** Specify the type of Encryption (AES 128/192/256).

---

**Encryption key (for Encryption enabled)** Click **Generate** to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.

---

**Recording parameters** Select this checkbox to customize the recording parameters.

---

**Destination Type** Select the radio button for the desired storage type and then fill in the relevant parameters.

- **Record to disk** – the recording is saved on the local disk.
  - **Record to S3** – the file is saved to your AWS S3 Bucket.
- 

#### **Record to Disk Parameters**

---

Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.

The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:

- File Name Template**
- %S=stream id - optional.
  - %Y=year - optional
  - %M=month - optional
  - %D=day - optional
  - %T=HH.MM.SS creation time - mandatory

For example, the following pattern %Y\_recording%T.ts will generate the following file name 2017\_recording15:32:35.ts.

---

**Max recorded file duration [hours]** Specify the maximum time allotted for recording a video stream. Default: 2 hours

---

## Configuring NDI Streams

---

### Adding an NDI Input to Zixi Broadcaster


---

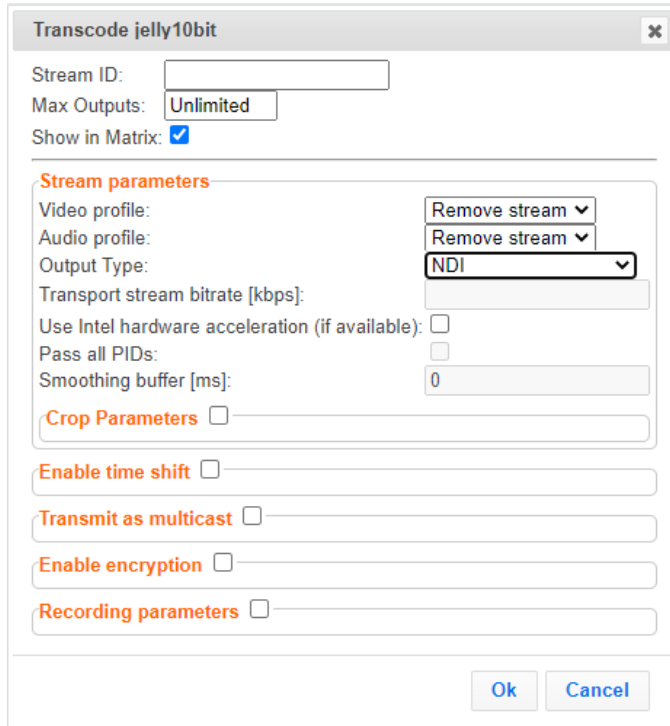
<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours
<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.
<b>Record to S3 Parameters</b>	
<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>URL</b>	Specify the URL of the S3 bucket.
<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.

## Creating an NDI Output

Zixi Broadcaster can transcode a TS stream Input into an NDI output using Raw video and audio profiles. The NDI output is configured in the transcoding profile itself, so there is no need to create an NDI Output from the Zixi Broadcaster Outputs screen. The Raw video profile can be configured to modify the stream's resolution (i.e. upscale/downscale) and frame rates per second. When the input stream is transcoded, it becomes available to be pulled by NDI enabled devices.

### ➔ To Create an NDI Output:

1. Create a Raw video transcoding profile and a Raw audio transcoding profile. For instructions, see [Creating New Profiles](#) section.
2. Create an input stream by following the instructions in the Adding Input Streams section.
3. Click  button on the right end of the input stream row and then **Transcode**. The **Transcode <input stream name>** window with the settings of the selected input stream is displayed.
4. Enter the unique **Stream ID.**, which will correspond to the new transcoded input that will be added.
5. Under **Stream Parameters**, in the **Output Type** field, select **NDI**.



6. Under Stream Parameters, in the **Video profile** and **Audio profile** fields, do one of the following:
  - Select the Raw Video profile and Raw Audio Profile that you have configured.

## Configuring NDI Streams

### Creating an NDI Output

- Select **Remove stream** – you can use the transcoder to remove the video or audio track from the stream.
7. Fill in additional fields (see table below for additional parameters).
  8. Click **OK**. The transcoded input stream is created and appears in an embedded window labeled Transcoded sources under the original input stream from which it was created.

Parameter	Description
<b>Video Profile</b>	Select an existing Raw Video profile, or select <b>Remove stream</b> if you want to remove the video track.
<b>Audio Profile</b>	Select an existing Raw Audio profile, or select <b>Remove stream</b> if you want to remove the audio track.
<b>Use Intel Hardware Acceleration (if available)</b>	Select this checkbox to use hardware acceleration component if it has been installed. When using an Nvidia profile, this setting will be disabled.
<b>Crop Parameters</b>	Select this checkbox to set the cropping parameters.
<b>Cropping Mode</b>	Set the units used for defining the crop. Options are: <i>Pixels</i> or <i>Percentages</i> . Note: Depending on the type of content, in some cases using different crop values for low resolution outputs might be appropriate. For example, if the content includes both video content and textual content, the text would not be readable at lower resolutions and thus might be better to exclude. Another use case for using different crop values for each output would be to slice a 360 or VR input video.
<b>Left, Top</b>	Specify the left and top positions for the crop.
<b>Width x Height</b>	Specify the width and height values for the crop.
<b>Enable time shift</b>	
<b>Maximum delay [hh:mm:ss]</b>	Specify the time of the delayed transmission of this stream. Maximum: 24 hours.
<b>Transmit as multicast</b>	Selecting this checkbox enables the transmission of this stream's outputs as multicast. A Zixi receiver that will pull the stream will receive it in multicast. By default, Zixi Broadcaster is configured to allow transparent fallback to unicast if it is out of the LAN. Requires enabling Multicast Pull in the Settings. This setting requires enabling Multicast Pool (Settings > Multicast Pool).
<b>Multicast Only</b>	Selecting this checkbox will force the transmission of this stream only in multicast.
<b>Enable Encryption</b>	Select this checkbox to encrypt the Input stream. For more information, see <a href="#">Encrypting an Input Stream in Zixi Broadcaster</a> .

## Configuring NDI Streams

---

### Creating an NDI Output

---

<b>Encryption type (for Encryption enabled)</b>	Specify the type of Encryption (AES 128/192/256).
<b>Encryption key (for encryption enabled)</b>	Click <b>Generate</b> to generate an encryption key. The generated encryption key must be sent to the end-user to decipher the received encoded stream.
<b>Recording parameters</b>	Select this checkbox to customize the recording parameters.
<b>Destination Type</b>	Select the radio button for the desired storage type and then fill in the relevant parameters. <ul style="list-style-type: none"><li>• <b>Record to disk</b> – the recording is saved on the local disk.</li><li>• <b>Record to S3</b> – the file is saved to your AWS S3 Bucket.</li></ul>
<b>Record to Disk Parameters</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>File Name Template</b>	
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>Keep recorded files for [hours]</b>	Specify the maximum time for storing the recording. Default: 0 hours
<b>Custom Path</b>	Optionally specify a storage location for the recorded files that is relative to the root folder. This location bypasses the location specified as the root folder in the <b>Settings &gt; General</b> screen.

## Configuring NDI Streams

---

### Creating an NDI Output

---

---

#### **Record to S3**

##### **Parameters**

---

<b>File Name Template</b>	<p>Specify the file name pattern that Zixi will use to generate when saving the recorded content. A separate file will be generated for each recording up to the time specified in the Max recorded file duration parameter (see above). After this duration a new file will be generated.</p> <p>The file name pattern can include any string and up to five variables listed below. Only the %T=HH.MM.SS creation time is mandatory. You can change the order of these variables and add text as long as you maintain the legitimate structure of the file in Linux and Windows:</p> <ul style="list-style-type: none"><li>• %S=stream id - optional.</li><li>• %Y=year - optional</li><li>• %M=month - optional</li><li>• %D=day - optional</li><li>• %T=HH.MM.SS creation time - mandatory</li></ul> <p>For example, the following pattern %Y_recording%T.ts will generate the following file name 2017_recording15:32:35.ts.</p>
<b>Max recorded file duration [hours]</b>	Specify the maximum time allotted for recording a video stream. Default: 2 hours
<b>URL</b>	Specify the URL of the S3 bucket.
<b>Ignore TLS certificate errors</b>	TLS certificate for S3 bucket might be detected as faulty if bucket name contains dot[s]. When this option is selected, Zixi Broadcaster will ignore the TLS certificate errors and transmit the stream to its destination.
<b>Access Key</b>	The access key that is used for accessing the S3 bucket.
<b>Secret Key</b>	The secret key that is used for accessing the S3 bucket.

---



# Configuring WebRTC Streams

Zixi supports using WebRTC to monitor streams. With WebRTC you can monitor your streams using popular browsers, such as Chrome, without installing any external plugins, such as the Zixi VLC plugin. This guide described the process of transcoding a stream to support WebRTC and how to monitor a stream using the WebRTC browser-integrated player.

## Video and Audio Codec Support Options

WebRTC stream delivery requires particular audio and video codecs. In case the streams you are transporting via Zixi Broadcaster do not match these requirements, additional transcoding will be necessary in order to re-encode the stream to become WebRTC compliant.

The following are the requirements for a Zixi stream to be *WebRTC compliant*:

- **Video** - H.264 codec with Baseline or Main profile and no B frames
- **Audio** – Opus codec

The following is a description of the possible methods for processing streams in Zixi for WebRTC viewing:

- **Pass-Through** – if the stream is already fully compliant with WebRTC, then it can be passed through the Zixi Broadcaster to the WebRTC player for viewing.
- **Transcoding** – if the incoming stream is not fully compliant, then you need to first transcode the stream in order to prepare it for delivery to the WebRTC player. There are two methods that can be used for transcoding a stream for WebRTC.
  - **Single Version - Creating a single version of the monitor stream for all users** – in this option the stream will be encoded using Zixi Broadcaster's transcoder (with existing X264 or Nvidia for transcoding). This option allows many users to monitor the stream without encoding for each individual user, which saves CPU capacity. However, the option will still require additional output bandwidth for each user. Since there is only one stream version, it is recommended to encode to a low resolution and low bitrate so that the stream will be viewable by all users monitoring the stream.
  - **Multiple Version - Creating a unique version of the monitor stream for each user** – in this option Zixi Broadcaster will transcode the video, transforming it into RAW frames, which will be passed to the WebRTC library, where it will be re-encoded for each individual user. In this option the stream will match the bitrate available for every particular user. Although this promises the best quality for users' specific network conditions, each re-encode consumes CPU, which

## Configuring WebRTC Streams

### Video and Audio Codec Support Options

could potentially max out the available CPU resources used to transport the streams. In this case, if the stream will be monitored by a large number of users, it is recommended to host a second Broadcaster and send the stream to the second Broadcaster and have all users monitor the stream from that second Broadcaster. By doing so, there will be no impact on the primary Broadcaster.

Input Codec	Passthrough	Single Version	Multiple Version
<b>Video Codecs</b>			
<b>H.264 with no B-frames (profile Baseline/Main)</b>	Yes	Not Relevant	Not Relevant
	No	Transcode to WebRTC compliant profile (e.g. H.264 codec with Baseline or Main profile and no B frames). Needs transcoding on a server with Nvidia GPUs.	Transcode to a RAW profile and specify a low resolution.
<b>H.264 with B-frames</b>	No	Transcode to WebRTC compliant profile. Needs transcoding on a server with Nvidia GPUs.	Transcode to a RAW profile and specify a low resolution.
<b>MPEG2</b>	No	Transcode to WebRTC compliant profile. Needs transcoding on a server with Nvidia GPUs.	Transcode to a RAW profile and specify a low resolution.
<b>HEVC</b>	No	Transcode to WebRTC compliant profile. Needs transcoding on a server with Nvidia GPUs.	Transcode to a RAW profile and specify a low resolution.
<b>Audio Codecs</b>			
<b>Opus</b>	Yes	Not Relevant	Not Relevant
<b>AAC</b>	No	Transcode to Opus codec.	Transcode to RAW.
<b>AC-3LC / HE v1 / HE v2</b>	No	Transcode to Opus codec.	Transcode to RAW.
<b>MPEG-2 Audio Layer II</b>	No	Transcode to Opus codec.	Transcode to RAW.

## Configuring WebRTC Streams

---

### Video and Audio Codec Support Options

---

---

<b>MPEG-1 Audio Layer II</b>	No	Transcode to Opus codec.	Transcode to RAW.
<b>E-AC-3 (and other unsupported codecs)</b>	Not supported. The stream will be passed through to WebRTC without audio.	Not relevant	Not relevant

---

---

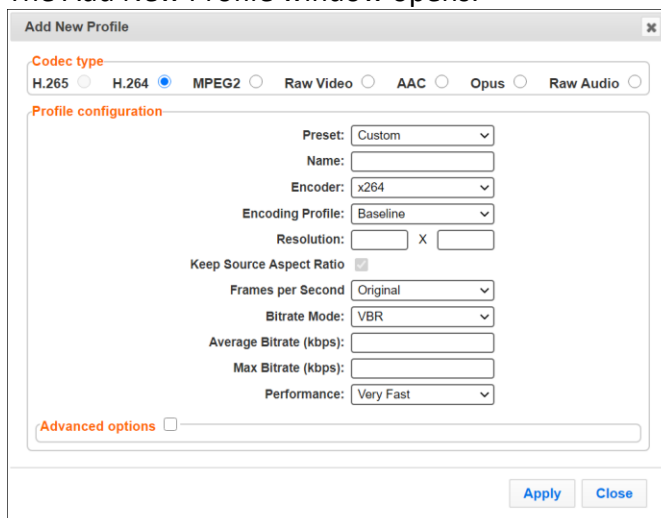
## Creating Transcoding Profiles for WebRTC

In order to enable transcoding of non-WebRTC compliant streams you need to first create the relevant Transcoding Profiles. You need to create separate video and audio profiles for use with Single Version and Multiple Version transcoding methods. You will then be able to apply those profiles to your streams as needed.

### Creating a Video Profile for “Single Version” Transcoding

#### ➔ To create a video profile for “Single Version” transcoding:

1. On the **Transcoder** screen of the Broadcaster UI, click **New Profile**. The Add New Profile window opens.



2. Under **Codec type**, select the **H.264** radio button (default).
3. In the **Name** field, enter a name that describes this profile (e.g. WebRTCSingleVersionVideo)
4. In the **Encoding Profile** field, select either *Baseline* or *Main*.
5. In the **Resolution** fields, enter a low resolution that can be viewed by all of your users.
6. Configure additional parameters as required, see [Creating New Profiles](#).
7. Click **Apply**.  
The new profile is created and it can be used to configure a “Single Version” WebRTC output.

### Creating an Audio Profile for “Single Version” Transcoding

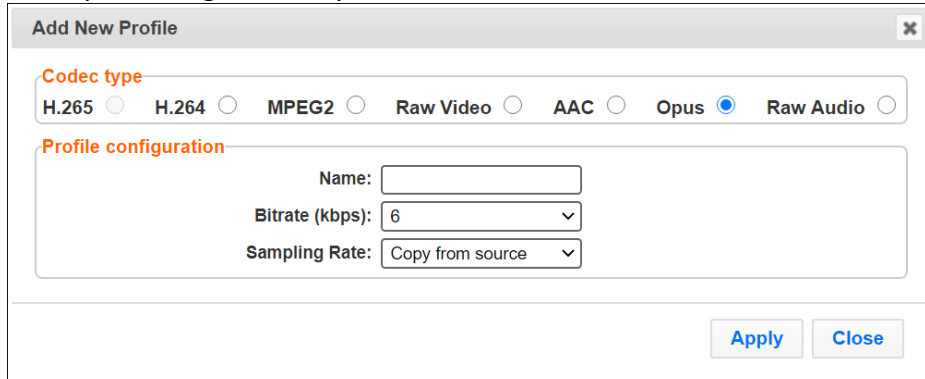
#### ➔ To create an audio profile for “Single Version” transcoding:

1. On the **Transcoder** screen of the Broadcaster UI, click **New Profile**. The **Add New Profile** window opens.

## Configuring WebRTC Streams

### Creating Transcoding Profiles for WebRTC

- Under **Codec type**, select the **Opus** radio button (default).  
The Opus configuration options are shown.



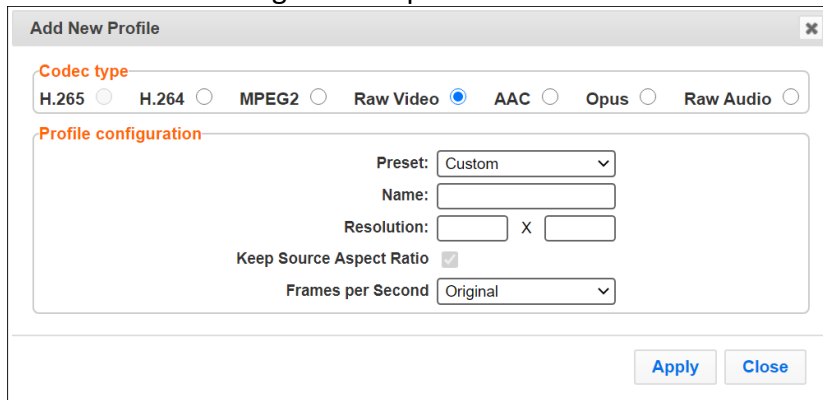
The screenshot shows the 'Add New Profile' dialog box. Under the 'Codec type' section, the 'Opus' radio button is selected. The 'Profile configuration' section contains three fields: 'Name' (a text input field), 'Bitrate (kbps)' (a dropdown menu with '6' selected), and 'Sampling Rate' (a dropdown menu with 'Copy from source' selected). At the bottom right, there are 'Apply' and 'Close' buttons.

- In the **Name** field, enter a name that describes this profile (e.g. WebRTCSingleVersionAudio)
- Configure additional parameters as required, see [Creating New Profiles](#).
- Click **Apply**.  
The new profile is created and it can be used to configure a “Single Version” WebRTC output.

### Creating a Video Profile for “Multiple Version” Transcoding

#### ➔ To create a video profile for “Multiple Version” transcoding:

- On the **Transcoder** screen of the Broadcaster UI, click **New Profile**.  
The **Add New Profile** window opens.
- Under **Codec type**, select **Raw Video** (default).  
The Raw Video configuration options are shown.



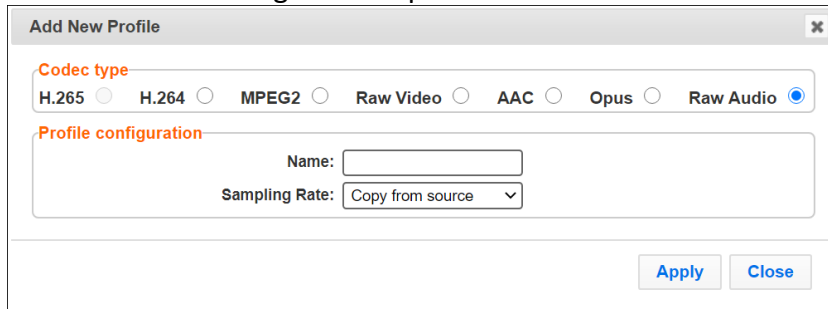
The screenshot shows the 'Add New Profile' dialog box. Under the 'Codec type' section, the 'Raw Video' radio button is selected. The 'Profile configuration' section contains five fields: 'Preset' (a dropdown menu with 'Custom' selected), 'Name' (a text input field), 'Resolution' (two text input fields separated by an 'X'), 'Keep Source Aspect Ratio' (a checked checkbox), and 'Frames per Second' (a dropdown menu with 'Original' selected). At the bottom right, there are 'Apply' and 'Close' buttons.

- In the **Name** field, enter a name that describes this profile (e.g. WebRTCMultipleVersionVideo)
- Configure additional parameters as required, see [Creating New Profiles](#).
- Click **Apply**.  
The new profile is created and it can be used to configure a “Multiple Version” WebRTC output.

## Creating an Audio Profile for “Multiple Version” Transcoding

### ➔ To create an audio profile for “Multiple Version” transcoding:

1. On the **Transcoder** screen of the Broadcaster UI, click **New Profile**.  
The Add New Profile window opens.
2. Under **Codec type**, select **Raw Audio** (default).  
The Raw Audio configuration options are shown.




3. In the **Name** field, enter a name that describes this profile (e.g. WebRTCMultipleVersionAudio)
4. Configure additional parameters as required, see [Creating New Profiles](#).
5. Click **Apply**.  
The new profile is created and it can be used to configure a “Multiple Version” WebRTC output.

## Creating a WebRTC Output

Zixi Broadcaster can transcode a TS stream Input into a WebRTC output using the relevant profiles, depending on the input video and audio, as described in the [Video and Audio Codec Support Options](#) section. Before creating the WebRTC output, make sure you have configured the relevant transcoding profiles [Creating Transcoding Profiles for WebRTC](#). When the input stream is transcoded, it becomes available to be pulled by the WebRTC player.

### ➔ To Create a WebRTC Output:

1. Create an input stream by following the instructions in the [Adding Input Streams](#) section of the Broadcaster UG.
2. Click  button on the right end of the input stream row, and then click **Transcode**. The **Transcode <input stream name>** window with the settings of the selected input stream is displayed.
3. Enter the unique **Stream ID.**, which will correspond to the new transcoded input that will be added.

## Configuring WebRTC Streams

### Creating a WebRTC Output

In the **Output Type** field, select **WebRTC**.

Transcode webrtc test

Stream ID:

Max Outputs:

Show in Matrix:

**Stream parameters**

Video profile:

Audio profile:

Output Type:

Transport stream bitrate [kbps]:

Use Intel hardware acceleration (if available):

Pass all PIDs:

Smoothing buffer [ms]:

Crop Parameters

Enable time shift

Transmit as multicast

Enable encryption

Recording parameters

Ok Cancel

4. Under **Stream Parameters**, in the **Video profile** field, select the desired video profile for the desired method.
  - a. Pass-Through – select **Keep original stream**
  - b. Single Version Transcoding – select a profile that converts to H.264 with Main or Baseline profile and no B Frames
  - c. Multiple Version Transcoding – select a profile that converts to RAW video
5. In the **Audio profile** field, select the desired audio profile for the desired method.
  - d. Pass-Through – select **Keep original stream**
  - e. Single Version Transcoding – select a profile that converts to Opus
  - f. Multiple Version Transcoding – select a profile that converts to RAW audio




Only supported profiles are shown. If you would like to remove the video or audio stream completely, select **Remove stream**.

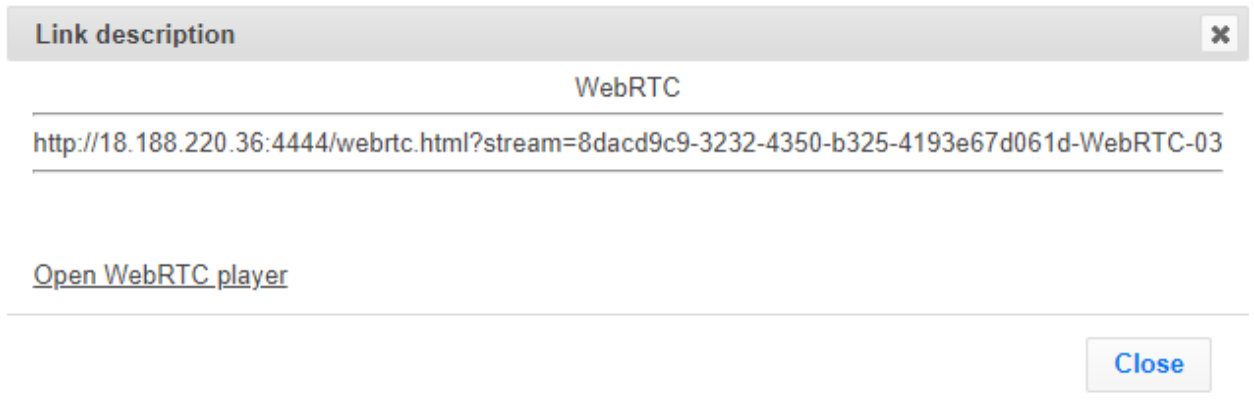
6. Configure additional parameters as required, see [Transcoding an Input Stream](#).
7. Click **OK**. The transcoded input stream is created and appears in an embedded window labeled Transcoded sources under the original input stream from which it was created.

## Playing and input stream with WebRTC player

WebRTC uses a regular browser to play the stream without the need to download and install an external plug in.

### ➔ To Play Input Streams with WebRTC:

1. In the Zixi Broadcaster **Inputs** page, click the desired WebRTC input stream that you want to play.
2. Click the  button on the right end of the selected row and then **Play WebRTC**. The **Link Description** window will open with the directory path for the desired file and a link to open a browser window to play the stream.



3. Copy the link or click the **Open WebRTC player** link to view the stream.



# Configuring MultiPath TCP

Developed by the Internet Engineering Task Force's (IETF) Multipath TCP working group, MultiPath TCP (MPTCP) is an effort towards enabling the simultaneous use of several IP-addresses/interfaces by a modification of TCP that presents a regular TCP interface to applications, while in fact spreading data across several subflows. This means that the Linux Kernel should be updated to support MPTCP.

The MPTCP functionality is similar to Zixi's "multi-link bonding" feature, which is available for UDP only, while MPTCP works on TCP.

If both the sending end and the receiving end support MultiPath TCP, Zixi can use this functionality to transmit a stream over TCP using multiple network interfaces (e.g. Wifi + mobile). The multiple links can be configured to:

- **Aggregate** - aggregates the bandwidth of the multiple links in order to use the total capacity of these links. In this case Zixi will use all available bandwidth among all available flows (if needed) to send the stream.
- **Duplicate** - enhance the path's resiliency by duplicating the stream and sending through all links in order to overcome errors and packet loss.

## MultiPath Modes

MultiPath TCP can be used in two modes:

- **Pull** – when pulling the stream, the receiving-end has multiple network interfaces (e.g. two NIC) while the sending side has one network interface. When the stream is pulled by the receiving-end using the MultiPath feature, the client (receiving side / puller) creates the sub-flows, (one per nic) and connects to the "source" (the server). The definition of the NICs in the destination is done automatically at the OS socket level. To configure this mode:
  - **In the sending side** - In **Settings > Live Protocols** screen, enable the **Allow automatic Pull output option**. In this option the output will be created automatically when the Pull connection is established. Following the connection, the two links will be displayed as bonded links. When the connection is established, the two links will be displayed as bonded links as shown below.

## Configuring MultiPath TCP

### MultiPath Modes

Status	Name	Type	Destination	Bitrate[kbps]	Uptime	Latency	Input ID
Connected	HTTP_10.6.0.17314823...	HTTP	169.1.119.188:44750	1128	00:00:04	0	aaa

Status	Link	Bitrate[kbps]	Up Time
Connected	41.13.96.197:56207@10.6.0.173	0	00:00:02
Connected	169.1.119.188:52910@10.6.0.173	1180	00:00:02

- **In the receiving side** – Configure a **TS over HTTP Input as Pull**. When the connection is established, only one link is displayed as shown below.

Status	ID	Type	Source	Bitrate[kbps]	Up Time	TR 101 290	Error
Offline	aaa	File	20.ts	0	00:00:00	Off	Stopped
Connected	ts-pull	HTTP	http://52.91.24.109:7777/aaa.ts	1149	00:00:08	Off	None

- **Push** – when pushing the stream, the receiving-end has one or more network interfaces (e.g. two NICs) while the sending side can have one network interface. In this case, you should configure it in the following way:

- **In the sending side** – you will need to configure a **TCP output** from the Zixi Broadcaster in the sending side. In this case you can configure the two links as **Aggregate** or **Duplicate**, as described above. When the connection is established, the two links will be displayed as bonded links as shown below.

Status	Name	Type	Destination	Bitrate[kbps]	Uptime	Latency	Input ID
Stopped		UDP	10.0.0.37:10000	0	00:00:00	0	8000
Stopped		UDP	127.0.0.1:8000	0	00:00:00	0	aaa
Connected	bbb	TCP	52.91.24.109:7777	1113	00:01:18	0	aaa

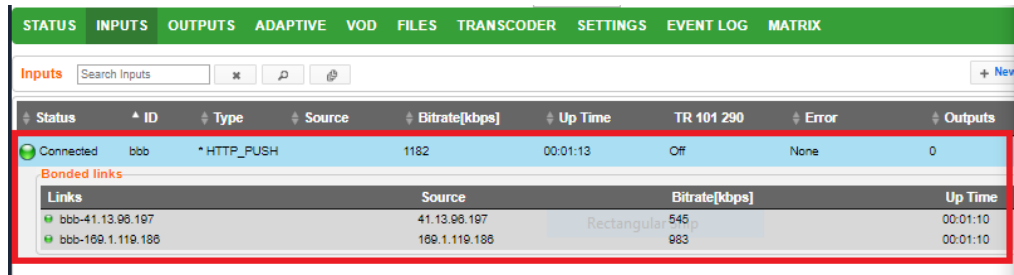
  

Status	Link	Bitrate[kbps]	Up Time
Connected	52.91.24.109:7777@172.20.10.10	402	00:01:17
Connected	52.91.24.109:7777@10.0.0.75	1058	00:01:17

- **In the receiving side** - In the receiving-end, if this is also a Zixi Broadcaster, you have two options:
  - Configure a **TS over HTTP Input as Push**. In this option the Push input is ready in advance and will show the bitrates of both TCP connections when the connection has been established.
  - In **Settings > Live Protocols** screen, enable the **Allow automatic Push input option**. In this option the input will be created automatically when the Push connection is established. When the connection is established, the two links will be displayed as bonded links as shown below.

## Configuring MultiPath TCP

### MultiPath TCP Setup



Status	ID	Type	Source	Bitrate[kbps]	Up Time	TR 101 290	Error	Outputs
Connected	bbb	* HTTP_PUSH		1182	00:01:13	Off	None	0

Links	Source	Bitrate[kbps]	Up Time
bbb-41.13.98.197	41.13.98.197	545	00:01:10
bbb-109.1.119.188	109.1.119.188	983	00:01:10

## MultiPath TCP Setup

Zixi supports MultiPath TCP for Linux based on the OS Kernel support. This means that the Linux Kernel should be updated to support MPTCP.

### ➔ To setup MultiPath TCP:

- Follow the relevant instructions in - <https://multipath-tcp.org/pmwiki.php/Users/HowToInstallMPTCP?>

# Using an External Storage

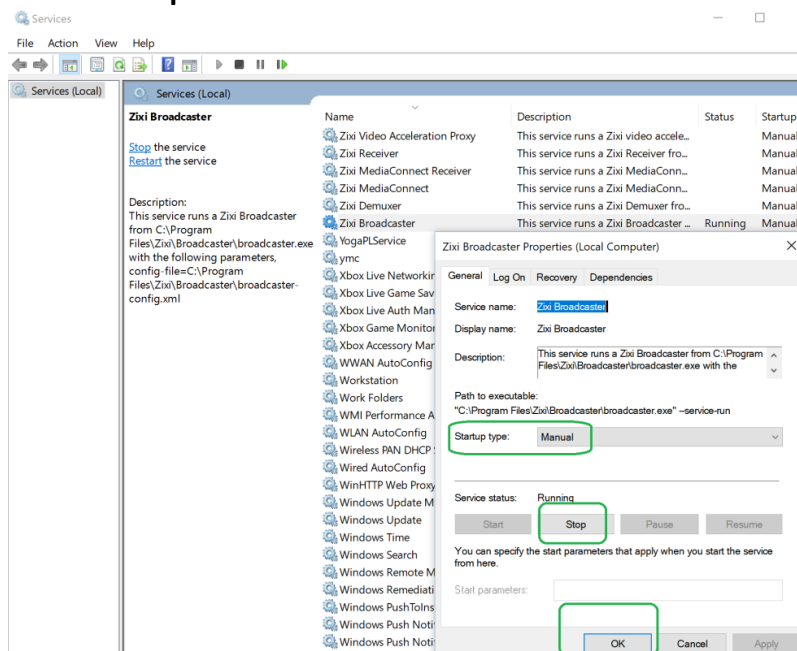
Modification of location of the Zixi Broadcasters 'cache' directory may be desired to minimize HDD used at the default location for recording streams. The default storage location (C:\Program Files\Zixi\Broadcaster\cache) can be replaced by using a symbolic link.



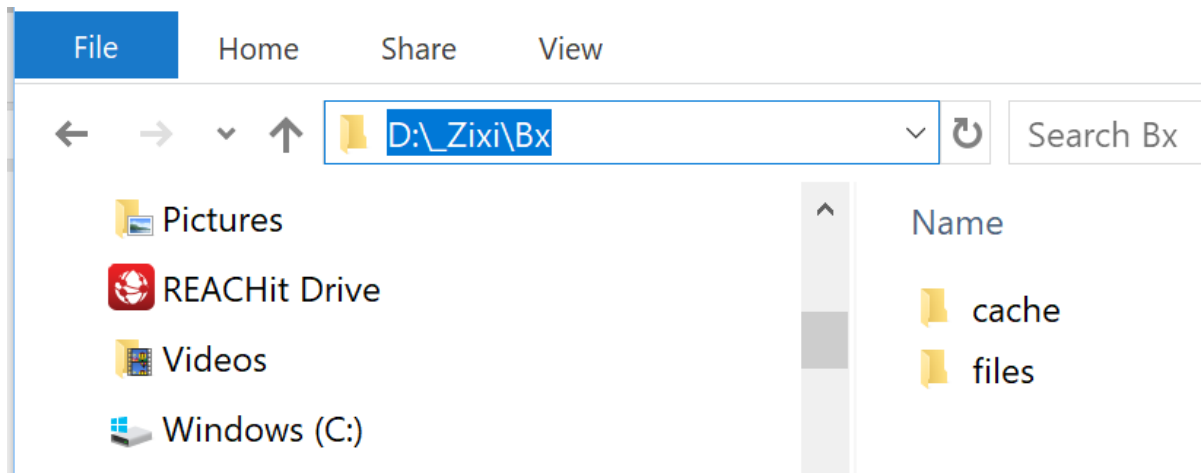
The Zixi Broadcaster will need to be in a Maintenance Window to apply these steps as the service will be turned off and on.

## ➔ To configure an external storage path in windows:

1. On the Zixi Broadcaster Windows machine, stop the Zixi Broadcaster service, by going into the operating system **Services** screen.
2. Double click the Zixi Broadcaster service to open its configuration properties.
3. Under the **General** tab, in the **Startup type** drop-down menu, select the **Manual** option.
4. Click the **Stop** button and then click **OK**.



5. Delete the 'cache' file located at C:\Program Files\Zixi\Broadcaster\cache. This may require Administrator privileges depending on your environment.
6. On the planned new location create your new cache directory. In the example below the directory was created in d:\\_Zixi\Bx\cache.



7. Go into the Windows Command line (cmd) as an Administrator.
8. Enter a makelink command to create a new Symbolic link from the original path to the newly created 'cache' folder. For example:

```
mklink /D "c:\Program Files\Zixi\Broadcaster\cache" d:\_Zixi\Bx\cache
```

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17134.885]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>mklink /D "c:\Program Files\Zixi\Broadcaster\cache" d:\_Zixi\Bx\cache
symbolic link created for c:\Program Files\Zixi\Broadcaster\cache <=> d:\_Zixi\Bx\cache

C:\WINDOWS\system32>
```

The newly created symbolic link for cache will be displayed in the following directory -  
C:\Program Files\Zixi\Broadcaster\

9. Go into the operating system **Services** screen.
10. Double click the Zixi Broadcaster service to open its configuration properties.
11. Under the **General** tab, in the **Startup type** drop-down menu, select the Automatic option.
12. Click the **Start** button and then click **OK**.

The recording of input streams will be stored in the new directory.

---

-----  
- License for OpenSSL  
-----

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit

(<http://www.openssl.org/>)

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim

Hudson (tjh@cryptsoft.com).

Copyright (c) 1998-2017 The OpenSSL Project. All rights reserved.

-----  
- License for Libssh2  
-----

Copyright (c) 2004-2007 Sara Golemon <sarag@libssh2.org>

Copyright (c) 2005,2006 Mikhail Gusarov <dottedmag@dottedmag.net>

Copyright (c) 2006-2007 The Written Word, Inc.

Copyright (c) 2007 Eli Fant <elifantu@mail.ru>

Copyright (c) 2009-2014 Daniel Stenberg

Copyright (C) 2008, 2009 Simon Josefsson

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the copyright holder nor the names of any other contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

-----  
- License for DekTec's Linux SDK. Version 1, August 2012  
-----

Copyright (C) 2000-2015 DekTec Digital Video B.V.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary format must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation.

THIS SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL DEKTEC DIGITAL VIDEO BV, ITS AGENTS OR ITS EMPLOYEES BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR OTHER DAMAGES (INCLUDING DAMAGES FOR THE LOSS OF USE, INFORMATION, GOODWILL, PROFIT, WORK STOPPAGE, DATA, BUSINESS OR REVENUE) UNDER ANY CIRCUMSTANCES, OR UNDER ANY LEGAL THEORY, WHETHER IN CONTRACT, IN TORT, IN NEGLIGENCE, OR OTHERWISE, ARISING FROM THE USE OF, OR INABILITY TO USE THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.